



UNIVERSIDADE
LUSÓFONA

iEquus

Trabalho Final de Curso

Relatório

Duarte Chen • a22203153

Professor Doutor Daniel Fernandes
Professor Doutor João Carvalho

Trabalho Final de Curso | LEI | 27/06/2025

www.ulusofona.pt

Direitos de cópia

iEquus, Copyright de Duarte Nuno Silva Chen, Universidade Lusófona.

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona (UL) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Este documento foi gerado com o processador (pdf/Xe/Lua) \LaTeX e o modelo ULThesis (v1.0.0) [Mat24].

Resumo

A evolução tecnológica está a transformar a saúde animal, incluindo a prática veterinária equina, onde a precisão na avaliação do BCS (*Body Condition Score*) e do BW (*Body Weight*) dos cavalos é essencial para o seu bem-estar e desempenho. Estudos como o de Urbanek e Zebeli (2023) destacam a importância de uma abordagem mais detalhada para estimar o BCS e o BW, incorporando novos indicadores de musculatura e gordura localizada.

Este trabalho propõe o desenvolvimento de uma aplicação móvel, baseada na *framework* Flutter, que utiliza algoritmos de processamento de imagem e de *machine learning* para prever o BCS e BW de cavalos a partir de fotografias. A aplicação permite obter estimativas através de medições visuais, automatizando e simplificando o processo de monitorização do animal a partir destas métricas. Para isso, foi desenvolvida uma solução composta por dois servidores em Flask, uma base de dados MySQL e uma aplicação móvel desenvolvida em Flutter.

O público-alvo desta aplicação são estudantes de Medicina Veterinária da Universidade Lusófona e o Hospital Veterinário de Equinos de Santo Estêvão, proporcionando-lhes uma ferramenta prática que complementa a aprendizagem clínica e fomenta a familiaridade com tecnologias emergentes. Em parceria com a Faculdade de Medicina Veterinária, a aplicação visa não só melhorar a avaliação de BCS e BW, mas também servir de base para futuras pesquisas e integrar-se ao mercado de cuidados de saúde equina, preenchendo lacunas práticas e tecnológicas na área.

Palavras-Chave: tecnologia, saúde animal, *Body Condition Score* (BCS), *Body Weight* (BW), processamento de imagem, *machine learning*, aplicação móvel, saúde equina, monitorização animal, avaliação corporal automatizada, Flutter.

Abstract

Technological developments are transforming animal health, including equine veterinary practice, where accurate assessment of the body condition score (BCS) and body weight (BW) of horses is essential for their welfare and performance. Studies such as that of Urbanek and Zebeli (2023) highlight the importance of a more detailed approach to estimating BCS and BW, incorporating new indicators of musculature and regional fat.

This thesis proposes the development of a mobile application based on the Flutter framework, which uses image processing and machine learning algorithms to predict BCS and BW of horses from photographs. The application makes it possible to obtain estimates through visual measurements, automating and simplifying the process of monitoring the animal based on these metrics. To achieve this, a solution was developed consisting of two Flask servers, a MySQL database, and a mobile application built with Flutter.

The app is targeted at veterinary students at the University of Lusófona and professionals of the Santo Estêvão Equine Veterinary Hospital, providing them with a practical tool that complements clinical learning and promotes familiarity with new technologies. In partnership with the Faculty of Veterinary Medicine, the app aims not only to improve the assessment of BCS and BW, but also to serve as a basis for future research and integration into the equine healthcare market, filling practical and technological gaps in the field.

Keywords: technology, animal health, Body Condition Score (BCS), Body Weight (BW), image processing, machine learning, mobile application, equine health, animal monitoring, automated body assessment.

Conteúdo

Resumo	1
Abstract	3
Conteúdo	4
Lista de Figuras	6
Lista de Tabelas	7
1 Introdução	8
1.1 Enquadramento	8
1.2 Motivação e Identificação do Problema	8
1.3 Objetivos	9
1.4 Estrutura do Documento	9
2 Pertinência e Viabilidade	11
2.1 Pertinência e Relevância do Projeto	11
2.1.1 Impacto na Prática Veterinária	11
2.1.2 Apoio à Gestão de Saúde Equina	11
2.2 Viabilidade	11
2.2.1 Viabilidade Social	11
2.2.2 Viabilidade Tecnológica	12
2.2.3 Viabilidade de Continuidade	12
3 Benchmarking	13
3.1 Concorrência	13
3.1.1 Barkyn Assistant	13
3.1.2 Equine Weight Management	14
3.1.3 EquiTrace	14
3.1.4 Sleip	15
3.1.5 Horse Scanner	15
3.2 Proposta de inovação e mais-valias	16
3.3 Análise SWOT	17
4 Especificação e Modelação	18
4.1 Análise de Requisitos	18
4.1.1 Enumeração de Requisitos	18
4.1.2 Descrição detalhada dos requisitos principais	22
4.1.3 Use Cases	24
4.2 Modelação	24
4.3 Protótipos de Interface	28
5 Solução	30
5.1 Front-end	31
5.1.1 Flutter (Dart)	31
5.2 Back-end	31

5.2.1	Flask (Python)	31
5.2.2	Algoritmo de <i>Machine Learning</i>	32
5.2.3	MySQL (Base de Dados)	32
5.3	Arquitetura da Solução	33
5.4	Solução desenvolvida	33
6	Calendário	34
7	Plano de Testes e Validação	35
8	Resultados	37
8.1	<i>Update</i> : 27 de abril de 2025	37
8.1.1	Progresso de trabalho dos requisitos	37
8.1.2	Screenshots do <i>front-end</i>	42
8.2	<i>Update</i> : 27 de junho de 2025	46
8.2.1	Progresso de trabalho dos requisitos	47
8.2.2	Screenshots do <i>front-end</i>	50
9	Conclusão	52
	Bibliografia	54
	Anexos	55
9.1	Questionário de Satisfação	56
9.2	Manual de Instalação em Android	59
	Glossário	61

Lista de Figuras

1	Barkyn [Bar]	13
2	Equine Weight Management [Gla]	14
3	EquiTrace [Ltd]	14
4	Sleip [AB]	15
5	Horse Scanner [Gmb]	15
6	Diagrama de <i>Use Cases</i>	25
7	Diagrama entidade-relação na terceira forma normal	26
8	Protótipos dos ecrãs de <i>Login</i> e <i>Home</i>	28
9	Protótipos dos ecrãs da lista de cavalos	29
10	Protótipos dos ecrãs da lista de clientes	29
11	Arquitetura da Solução	30
12	Cronograma, de alto nível, das atividades do projeto	34
13	<i>Screenshots</i> dos ecrãs de <i>login</i> , da <i>dashboard</i> , da lista e perfil de cavalos .	42
14	<i>Screenshots</i> dos ecrãs para a criação de uma medição de BW e BCS . .	43
15	<i>Screenshots</i> dos ecrãs com o resultado da marcação das coordenadas e o BW e BCS previstos pela API	44
16	<i>Screenshots</i> dos ecrãs do raio-x	45
17	<i>Screenshots</i> do ecrã de consultas	46
18	<i>Screenshots</i> do ecrã de perfil de cavalos	50
19	<i>Screenshots</i> dos ecrãs de <i>login</i> e registo	51
20	<i>Screenshots</i> dos ecrãs de <i>login</i> e registo	51

Lista de Tabelas

1	Comparação entre soluções para gestão de saúde equina	16
2	Requisitos Funcionais	21
3	Requisitos Não Funcionais	22
4	Requisitos Técnicos	22
5	Tabela de Testes	36
6	Resultados: Requisitos Funcionais	39
7	Resultados: Requisitos Funcionais	49

1 - Introdução

“We can learn a lot from global health about how to make AI more equitable. The main lesson is that the product must be tailored to the people who will use it.” - Bill Gates, 2024. [Gat]

O setor da saúde animal, tal como muitos outros setores, está em constante evolução, impulsionado por inovações tecnológicas que prometem transformar a forma como monitorizamos e tratamos os animais. A utilização de ferramentas digitais, que utilizam algoritmos de inteligência artificial, tem-se demonstrado ser cada vez mais eficaz na melhoria dos cuidados prestados em qualquer área da saúde. Para os veterinários, o uso destas soluções significa não só o acesso a dados com maior facilidade, mas também a possibilidade de os aprofundar, garantindo que as suas decisões são fundamentadas com as melhores informações possíveis. No setor equino, métricas como o *Body Condition Score* (BCS) e o *Body Weight* (BW) são críticas para medir a saúde geral do cavalo e a sua precisão é crucial. [UZ23]

1.1 Enquadramento

O estudo de Urbanek e Zebeli [UZ23] propõe uma metodologia para determinar o BCS que vai para além do sistema mais comumente utilizado, e desenvolvido por Don Henneke [HEN+83]. Enquanto este sistema se baseia em indicadores visuais e de palpação para recolher dados sobre a gordura em várias regiões do corpo do cavalo, o estudo de Urbanek e Zebeli considera um conjunto bastante mais abrangente de métricas, como a idade do cavalo, o comprimento do corpo, a circunferência do corpo, a altura ao garrote e a atrofia do pescoço. Estas métricas são mais precisas do que a palpação, embora exijam um grau de complexidade maior para serem realizadas.

Já no que se refere ao cálculo do BW, Don Henneke não estuda este indicador, enquanto que o estudo de Urbanek e Zebeli contempla a altura ao garrote, a circunferência do tórax, a circunferência da canela, o comprimento do corpo e a sua circunferência para o efeito. São também utilizados o BCS, o CNS (*Cresty Neck Score*) e a MASS (*Muscle Atrophy Scoring System*). [UZ23]

1.2 Motivação e Identificação do Problema

O presente trabalho surge da constatação de que há uma lacuna entre a prática clínica e a utilização de soluções tecnológicas acessíveis que ofereçam maior precisão e agilidade na avaliação do BCS e do BW. O BW é fundamental, não apenas como indicador da saúde e do envelhecimento de um cavalo, mas também para avaliar as suas necessidades de específicas de nutrição e treino. A procura de uma abordagem mais eficiente e automatizada na avaliação da saúde animal é impulsionada por uma maior consciencialização sobre o bem-estar animal. Proprietários e profissionais do setor equino procuram soluções que combinem precisão científica com facilidade de utilização, permitindo um acompanhamento contínuo e intervenções mais assertivas. Assim, o problema central a ser explorado é a complexidade e o tempo necessários para obter medições precisas de BCS e BW utilizando métodos manuais, que muitas vezes são limitados em termos de acessibilidade e repetibilidade. Tal representa um desafio, especialmente em situações

que exigem avaliações rápidas ou em regiões com acesso limitado a recursos veterinários especializados.

A metodologia proposta por Urbanek e Zebeli baseia-se na relação de diversas dimensões do cavalo, proporcionando uma previsão precisa do BCS, mas sobretudo do BW que revela um coeficiente de correlação de concordância de 0,97, exatidão a 0,99 e uma precisão de 0,97. A implementação tecnológica dos processos descritos neste estudo facilita a obtenção de resultados, otimizando o tempo e o esforço necessários para realizar todas as medições previstas no modelo do estudo. O estudo refere como limitação a subjetividade e a influência do viés do observador nos sistemas de pontuação para avaliar a atrofia muscular. Embora estes sistemas sejam úteis para indicar o grau de muscularidade e, consequentemente, BW e o BCS do animal, podem gerar variações nos resultados dependendo de quem realiza a avaliação. Tal compromete a consistência e a confiabilidade dos dados obtidos. Por isso, como sugestão de melhoria, o estudo aponta para a necessidade de métodos mais objetivos e quantitativos, como a análise por imagem, a histologia ou os marcadores moleculares, para reduzir as variações causadas por fatores humanos e produzir resultados mais fidedignos. [UZ23]

1.3 Objetivos

Para abordar esta questão, será desenvolvida uma aplicação móvel em *Flutter* que, a partir de uma fotografia de um cavalo, consiga fazer previsões precisas do BCS e BW, marcando pontos específicos no corpo do equino com recurso ao algoritmo de *machine learning*. Este algoritmo, fundamentado no estudo de Urbanek e Zebeli, foi desenvolvido pelos professores doutores Daniel Fernandes e João Carvalho, docentes da Universidade Lusófona. A aplicação calculará estes indicadores com base na relação entre as dimensões das linhas formadas pelos pontos assinalados na fotografia do cavalo. Deste modo, o objetivo é construir uma solução que melhore a prática clínica dos veterinários, facilitando o acompanhamento da saúde equina e tornando as avaliações do BCS e do BW mais acessíveis e automatizadas.

1.4 Estrutura do Documento

Este documento encontra-se estruturado pelas seguintes secções:

- Na Secção 1, é feita uma introdução à aplicação a desenvolver, enquadrando-a no estudo de Urbanek e Zebeli [UZ23] e definem-se os objetivos e a motivação subjacentes ao desenvolvimento deste TFC;
- Na Secção 2, avalia-se a pertinência e a viabilidade do desenvolvimento do iEquus, considerando diferentes impactos, bem como a sua viabilidade nos âmbitos social, tecnológico e de continuidade futura;
- Na Secção 3, procede-se a uma análise da concorrência, com enfoque em aplicações relacionadas com a saúde animal. Realiza-se também uma análise SWOT, para identificar os principais fatores internos e externos que poderão influenciar o impacto desta aplicação no mercado;
- Na Secção 4, são detalhadas as especificações e a modelação do projeto, incluindo a análise de requisitos funcionais, não funcionais e técnicos. Adicionalmente, são apresentados os casos de uso e a modelação do sistema através de um diagrama de entidade-relação na terceira forma normal. Esta secção também inclui os protótipos de interface;

- Na Secção ??, é descrita a solução proposta, com uma visão abrangente da arquitetura da aplicação, abordando o desenvolvimento do *back-end*, do *front-end* e a integração entre ambas as componentes;
- Na Secção 6, é apresentado o cronograma do projeto sob a forma de um diagrama de *Gantt*, que organiza temporalmente as tarefas a realizar e apoia o planeamento do trabalho;
- Na Secção 7, está descrito o plano de testes e validação que será executado pelo alunos da Faculdade de Medicina da Universidade Lusófona e pelos médicos Veterinários da clínica de Santo Estêvão;
- Na Secção 8, são apresentados os resultados das entregas programadas deste TFC;
- Na Secção 9, é apresentada a conclusão, onde se destacam as principais ideias e resultados alcançados no âmbito deste projeto até à primeira entrega intercalar;

2 - Pertinência e Viabilidade

Nesta secção, será analisada a relevância e a viabilidade do projeto iEquus, demonstrando como este se destaca como uma solução inovadora para a monitorização da saúde equina. Primeiramente, será apresentada a pertinência do projeto, realçando o impacto positivo que a sua aplicação terá na prática veterinária e na gestão da saúde equina. Em seguida, serão explorados os critérios de viabilidade social, tecnológica e de continuidade, que sustentam o potencial do iEquus para ultrapassar os limites de um projeto académico e consolidar-se como uma ferramenta prática, acessível e escalável no setor da saúde animal.

2.1 Pertinência e Relevância do Projeto

A pertinência do iEquus prende-se com o potencial para resolver desafios identificados pelo Doutor José Prazeres, docente e diretor clínico do Hospital Veterinário de Equinos de Santo Estêvão, no setor da saúde equina e melhorar os cuidados prestados aos animais. A solução proposta permite monitorizar a saúde dos cavalos com exatidão, colmatando as lacunas existentes nos métodos tradicionais de medição do BW e avaliação do BCS através da palpação e da observação visual.

2.1.1 Impacto na Prática Veterinária

O iEquus disponibilizará uma ferramenta que automatiza o cálculo de indicadores essenciais de saúde, como o BCS e o BW, utilizando imagens capturadas pelos próprios utilizadores da aplicação. A plataforma facilitará a tomada de decisões informadas sobre, por exemplo, a dieta, o treino e os cuidados médicos, proporcionando uma abordagem mais eficaz e eficiente para a gestão da saúde dos equídeos.

2.1.2 Apoio à Gestão de Saúde Equina

Além de calcular o BCS e o BW, a aplicação permitirá o armazenamento de um histórico completo da saúde dos cavalos, incluindo informações sobre consultas veterinárias, tratamentos e dados dos proprietários e cuidadores. Esta funcionalidade transforma o iEquus numa plataforma de gestão integral da saúde equina, permitindo aos profissionais de saúde monitorizar de forma contínua a evolução do estado de saúde dos animais. A centralização destas informações proporciona uma visão global da saúde do cavalo, facilitando a personalização de planos de cuidados a longo prazo e contribuindo para a prevenção de doenças.

2.2 Viabilidade

A viabilidade deste TFC é sustentada por uma análise abrangente que contempla os aspetos social, tecnológico e de continuidade, assegurando o seu potencial de sucesso e expansão do projeto a longo prazo.

2.2.1 Viabilidade Social

O setor da saúde animal está a atravessar um crescimento significativo, especialmente no que concerne à gestão da saúde equina, devido à crescente procura por so-

luções digitais [UZ23]. O iEquus insere-se neste panorama, proporcionando uma ferramenta que utiliza *machine learning* para calcular indicadores de saúde essenciais, como o BCS e o BW dos cavalos. A aplicação, compatível com os sistemas Android e iOS, facilita a sua adoção por veterinários, proprietários de cavalos e centros de treino, permitindo otimizar o acompanhamento da saúde animal de forma prática, eficiente, mas sobretudo abrangente. O iEquus está diretamente ligado à Faculdade de Medicina Veterinária da Universidade Lusófona, onde está a ser acompanhado em termos dos requisitos e contributos teóricos veterinários. Desta forma, assegura-se que o Hospital Veterinário de Equinos de Santo Estêvão utilizará esta aplicação no futuro.

2.2.2 Viabilidade Tecnológica

A viabilidade tecnológica do iEquus está salvaguardada pela utilização de tecnologias consagradas e amplamente reconhecidas. A aplicação é desenvolvida em Flutter, uma *framework* que facilita a otimização da aplicação para dispositivos móveis. A aplicação de *machine learning* permite analisar as imagens dos cavalos e calcular o BCS e o BW. O sistema de *back-end* utiliza o Flask, uma *framework* flexível e eficiente que suporta o processamento em tempo real das imagens e dos dados recolhidos, garantindo que a informação é organizada e acessível. A utilização de uma base de dados MySQL para armazenar os dados facilmente de forma segura e escalável é também um fator crucial para a viabilidade tecnológica da plataforma, assegurando que o sistema possa crescer e ser adaptado a novas funcionalidades sem comprometer o seu desempenho.

2.2.3 Viabilidade de Continuidade

O projeto foi concebido de forma a assegurar a sua continuidade e evolução. A arquitetura modular e escalável, baseada no uso de tecnologias como Flutter e Flask, permite implementar facilmente futuras melhorias e atualizações. A separação clara entre *front-end* e *back-end* facilita a implementação de novas funcionalidades, como a integração com dispositivos de monitorização de saúde em tempo real ou o registo de indicadores de saúde pelo veterinário, utilizando comandos de voz. A fase académica do projeto poderá ser seguida pelo desenvolvimento de uma versão comercial da aplicação, que pode ser mantida e melhorada continuamente, aproveitando os dados recolhidos e o *feedback* dos utilizadores da versão contemplada neste trabalho.

A viabilidade do projeto está sustentada pela combinação de tecnologias e um modelo de desenvolvimento que garante a continuidade do sistema a longo prazo. Para além disto, a pertinência e relevância do projeto são evidenciadas pela parceria com a Faculdade de Medicina Veterinária da Universidade Lusófona.

O iEquus oferece uma ferramenta que automatiza e otimiza a monitorização da saúde dos cavalos, contribuindo para um cuidado mais preciso, com benefícios tanto agora para os profissionais da área como para, no futuro, os proprietários de cavalos. Com estas características, a aplicação tem um grande potencial para se expandir e adaptar às necessidades do mercado, tornando-se uma solução de destaque no que se refere à saúde equina.

3 - Benchmarking

O mercado da saúde e gestão de cavalos tem registado um crescimento constante nos últimos anos, impulsionado por diversas tendências, nomeadamente a maior consciencialização sobre o bem-estar animal, o avanço tecnológico, e a utilização de soluções digitais para otimizar os cuidados e o acompanhamento dos animais. As soluções existentes no mercado enfatizam a gestão de cavalos, a monitorização de dados clínicos e a utilização de imagens, em fotografia ou vídeo, para avaliar ou detetar pormenores no cavalo, relacionados com a sua saúde ou com a identificação de raças.

No âmbito do projeto iEquus, foi realizado um *benchmark* das soluções já existentes no mercado com foco na saúde e monitorização de cavalos, bem como na identificação de traços de animais através de processamento de imagem. O objetivo deste estudo comparativo foi identificar diferentes funcionalidades e metodologias disponibilizadas por aplicações concorrentes e comprovar a elevada procura de soluções tecnológicas para a saúde equina. Para tal, avaliou-se em que medida cada uma destas soluções atende às necessidades do setor e como o iEquus pode diferenciar-se. Esta análise permite compreender o potencial inovador do iEquus e identificar as lacunas que esta aplicação poderá preencher com funcionalidades adicionais e uma abordagem mais integrada à saúde equina. Assim, foram analisadas as seguintes aplicações:

3.1 Concorrência

3.1.1 Barkyn Assistant

A Barkyn Assistant [Bar], representada na figura 1, apesar de ser uma aplicação focada na saúde de cães, permite avaliar a condição corporal, bucal, ocular e o pelo destes animais. As suas funcionalidades de processamento de imagem e a metodologia de cálculo da métrica de *Health Score* são relevantes para o iEquus, na medida em que ambas recorrem à utilização de fotografias para atribuir algum tipo de indicador do estado de saúde de um animal. No entanto, a Barkyn Assistant não contempla as especificidades da saúde equina, limitando-se à monitorização da saúde em cães.



Figura 1: Barkyn [Bar]

3.1.2 Equine Weight Management [Gla]

A aplicação Equine Weight Management [Gla], apresentada na figura 2, ajuda a monitorizar e a gerir o BCS de cavalos, utilizando o sistema introduzido por Don Henneke [HEN+83] que avalia diferentes partes do corpo do cavalo de 1 a 9 consoante a quantidade de gordura acumulada. Embora seja útil para monitorizar um dos parâmetros críticos da saúde equina, a aplicação não oferece um sistema de processamento de imagem para realizar o cálculo do BCS ou do BW, pelo que fica dependente da avaliação manual. Embora esta aplicação contemple características importantes, não permite ter uma visão holística da saúde do animal.

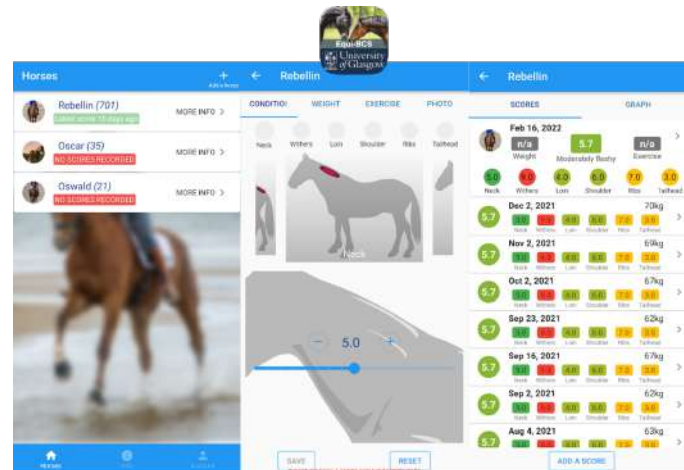


Figura 2: Equine Weight Management [Gla]

3.1.3 EquiTrace

A aplicação EquiTrace, [Ltd], ilustrada na figura 3, foca-se na gestão da saúde e identificação de cavalos, oferecendo funcionalidades úteis para criadores e veterinários. Liga-se a um microchip no cavalo, que regista a temperatura e a identidade do animal. Inicialmente, a aplicação limitava-se à ligação com o microchip, mas foi posteriormente alargada para incluir módulos de gestão de medicação e reprodução. No entanto, a solução não inclui um sistema de medição do BCS nem funcionalidades de processamento de imagem, limitando-se ao registo dos dados de saúde de cavalos.

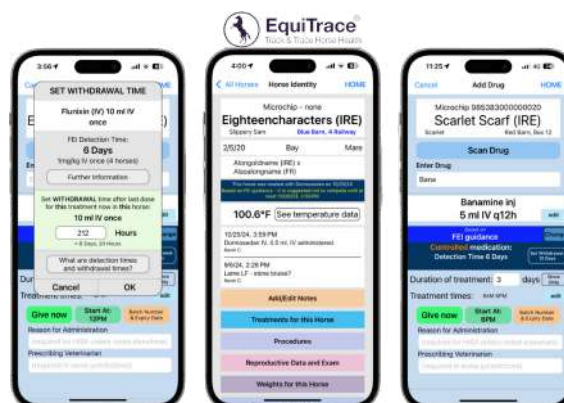


Figura 3: EquiTrace [Ltd]

3.1.4 Sleip

A aplicação Sleip [AB], representada na figura 4, analisa o movimento dos cavalos através do processamento de vídeos dos equinos em andamento, identificando assimetrias na sua marcha. Embora se destaque por ser uma aplicação que utiliza inteligência artificial, a Sleip foca-se exclusivamente na análise do movimento do equino, não oferecendo um sistema de BCS, nem de BW, nem uma avaliação abrangente do estado de saúde do cavalo.



Figura 4: Sleip [AB]

3.1.5 Horse Scanner

A aplicação Horse Scanner [Gmb], apresentada na figura 5, é uma solução que permite identificar a raça dos cavalos através de uma fotografia, com foco limitado a aspetos genéticos e de identificação visual. A aplicação não contempla a monitorização de saúde nem inclui um sistema de *Health Score* ou outras métricas relevantes para a gestão da saúde dos cavalos. Embora se foque na identificação das raças dos equinos, poderá ser relevante para o iEquus, na medida em que utiliza processamento de imagem, ainda que exclusivamente para identificação de raças.

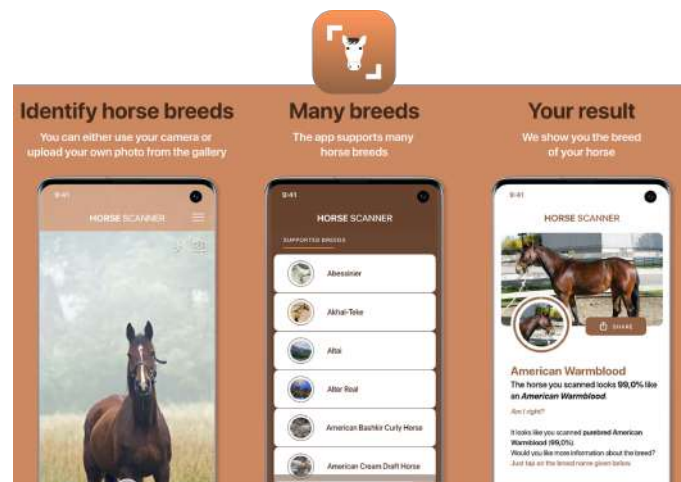


Figura 5: Horse Scanner [Gmb]

Solução	Android	iOS	Saúde	Registos	Processamento Imagem	Equinos
Barkyn	x	x	x	x	x	
Equine Weight Management	x			x		x
EquiTrace	x	x		x		x
Sleip		x		x	x	x
Horse Scanner	x	x			x	x

Tabela 1: Comparação entre soluções para gestão de saúde equina

As soluções analisadas e esquematizadas na tabela 1, apresentam abordagens distintas no âmbito da gestão de saúde animal, com diferentes focos e limitações, refletindo tanto as potencialidades como as lacunas do mercado atual. Entre estas, destaca-se a Barkyn Assistant [Bar], que, embora esteja orientada para cães, utiliza processamento de imagem para calcular indicadores de saúde, como o Health Score. Esta funcionalidade é relevante para o iEquus mas, a falta de adaptação às necessidades específicas da saúde equina limita a sua aplicabilidade direta. A Equine Weight Management [Gla] oferece uma solução prática para monitorizar o peso dos cavalos com base no sistema de avaliação BCS. No entanto, a ausência de processamento de imagem limita a automação e a precisão, obrigando a depender de medições manuais, o que restringe a visão global da saúde do animal. Por outro lado, a EquiTrace [Ltd] amplia as possibilidades de gestão ao integrar dados de saúde e identificação a utilização de microchips. No entanto, tal como a Equine Weight Management, não incorpora funcionalidades de BCS ou processamento de imagem, limitando-se ao registo de dados. Já a Sleip [AB] concentra-se exclusivamente na análise do movimento equino, utilizando inteligência artificial para identificar assimetrias na marcha. Apesar de ser uma ferramenta inovadora para a análise locomotora, não contempla a avaliação geral da saúde, como o BCS ou o peso. Por fim, o Horse Scanner [Gmb], com a sua capacidade de identificar raças através do processamento de imagem, também demonstra um foco muito específico, sem abordar métricas de saúde.

3.2 Proposta de inovação e mais-valias

A aplicação iEquus posiciona-se como uma solução inovadora e de grande relevância para o setor equino, proporcionando uma abordagem multidimensional que colmata as lacunas identificadas nas opções atualmente disponíveis. Com um modelo de *machine learning* para o cálculo do BCS e do BW através de fotografias, ferramentas de monitorização, funcionalidades de registo de saúde e gestão de cavalos, o iEquus responde à crescente necessidade de recursos capazes de assegurar um acompanhamento completo e eficiente do bem-estar e da saúde dos equinos.

A inovação do iEquus reside na sua capacidade de unificar diversos aspetos da monitorização da saúde equina numa plataforma única e especializada, ao contrário das soluções existentes, que se limitam a funcionalidades isoladas e pouco integradas. A aplicação diferencia-se pelas funcionalidades inéditas ou significativamente aprimoradas, com o cuidado de satisfazer as necessidades dos proprietários, mas indo mais além, ao oferecer suporte especializado a profissionais veterinários no tratamento de cavalos. Deste modo, possibilita uma análise criteriosa e detalhada das condições físicas e de saúde dos animais, com foco na monitorização do BW e do BCS, bem como de outros indicadores fundamentais, consolidando-se como uma ferramenta indispensável para o cuidado e gestão de equinos.

3.3 Análise SWOT

Adicionalmente, para melhor compreender o impacto real do desenvolvimento desta aplicação no mercado, realizou-se uma análise SWOT do mesmo:

Forças

- Inovação tecnológica;
- Fácil utilização, sem necessidade de balança para determinar o peso do cavalo;
- Solução acessível e menos invasiva para o animal;
- Redução de custos relativos a análises clínicas e exames veterinários;
- Promoção da independência do proprietário na monitorização contínua da saúde do animal;
- Contribuição para a construção de uma base de dados de saúde equina;
- Parceria com a Faculdade de Medicina Veterinária da Universidade Lusófona;
- Ferramenta de apoio clínico para veterinários.

Fraquezas

- Dependência da qualidade das fotografias;
- Dificuldade em obter dados precisos para estimar o BW e o BCS do cavalo a partir de fotografias;
- Mercado de nicho, o que limita o número potencial de utilizadores da aplicação;
- Resistência à mudança por parte de alguns proprietários e profissionais do setor.

Oportunidades

- Crescimento do mercado de aplicações voltadas para a saúde animal;
- Interesse crescente no cuidado, saúde e bem-estar dos animais;
- Potenciais parcerias com veterinários, clínicas veterinárias e centros de equitação.

Ameaças

- Forte concorrência no mercado de aplicações de saúde animal;
- Questões relacionadas com a regulamentação da informação utilizada e com a privacidade dos dados do utilizador e de saúde dos animais;
- Dependência de algoritmos de *machine learning* para a análise, o que pode comprometer a precisão em alguns casos;
- Adoção lenta da tecnologia, especialmente entre utilizadores que não estão familiarizados com inovações digitais.

4 - Especificação e Modelação

4.1 Análise de Requisitos

Para o projeto iEquus, foi elaborado um conjunto de requisitos funcionais, não funcionais e técnicos, presentes nas tabelas 2, 3 e 4, com o objetivo de atender às necessidades dos seus utilizadores, nomeadamente médicos veterinários, permitindo que estes cumpram os seus objetivos de forma eficiente.

4.1.1 Enumeração de Requisitos

Para avaliar a importância de cada requisito, foi usada a escala com os seguintes significados:

- nice to have: significa que o requisito é desejável, mas não essencial;
- should have: significa que o requisito é essencial;
- must have: significa que o requisito é imprescindível.

Para classificar o esforço necessário para o desenvolvimento de cada requisito, foi utilizada uma escala com base no número estimado de dias de trabalho e codificada como 'pequeno' (S), 'médio' (M) e 'grande' (L):

- S: para 1 a 2 dias de trabalho;
- M: para 3 a 5 dias de trabalho;
- L: para 6 ou mais dias de trabalho.

Requisitos Funcionais

ID	Requisito	Escala 'have'	Esforço
F1	Como médico veterinário, quero que depois de fotografar um cavalo possa verificar que a fotografia tem boa qualidade antes de a enviar para processamento.	must have	L
F1U1	Como médico veterinário, quero inserir fotografias da galeria para utilizar no processamento.	must have	S
F1U2	Como médico veterinário, quero que depois de tirar a fotografia decidir se pretendo mantê-la ou substituí-la.	must have	S

F1U3	Como médico veterinário, quero uma <i>frame</i> por cima da imagem da câmara com os contornos de um cavalo que me facilite a posicionar o <i>frame</i> da fotografia em relação ao cavalo.	nice to have	S
F1U4	Como médico veterinário, quero um tutorial para me orientar no início do processo de tirar uma fotografia.	should have	S
F1U5	Como médico veterinário, quero um botão que permita ligar o flash do meu dispositivo móvel para tirar uma fotografia em condições de baixa luminosidade.	must have	S
F1U6	Como médico veterinário, quero um botão que me permita sair do ecrã da câmara e regressar ao ecrã anterior.	must have	S
F2	Como médico veterinário, quero ver o resultado do processamento, o BW e o BCS previstos, e a fotografia com os pontos marcados pelo algoritmo.	must have	L
F2U1	Como médico veterinário, quero poder escrever o BW e o BCS reais de modo a alimentar o dataset do algoritmo de <i>machine learning</i> e manter um histórico do cavalo corretamente atualizado.	must have	M
F2U2	Como médico veterinário, quero rever a fotografia utilizada com os pontos marcados de modo a validá-los.	must have	S
F2U3	Como médico veterinário, quero saber quando é que os resultados estão prontos e quando é que estão em processamento.	must have	S
F2U4	Como médico veterinário, quero um histórico de resultados anteriores, para comparar o progresso e avaliar mudanças no estado de saúde do cavalo ao longo do tempo.	must have	L
F2U5	Como médico veterinário, quero poder fazer zoom e navegar na imagem processada, para visualizá-la e verificar a precisão dos pontos marcados.	should have	S
F3	Como médico veterinário, quero editar a posição dos pontos marcados na fotografia e enviar para processamento para o algoritmo de <i>machine learning</i> .	must have	L

F3U1	Como médico veterinário, quero ver um tutorial de como devem ser marcados na fotografia os pontos necessários, antes de os ter de marcar.	nice to have	L
F3U2	Como médico veterinário, quero poder fazer zoom e navegar pela a imagem enquanto marco os pontos, para garantir a maior precisão.	nice to have	M
F3U3	Como médico veterinário, quero que os pontos em cima da fotografia estejam com cores diferentes para cada objetivo metodológico, para arrastá-los pela imagem e marcar os locais fundamentais ao processamento da imagem do cavalo.	must have	S
F3U4	Como médico veterinário, quero poder desfazer ou refazer as movimentações recentes dos pontos na imagem, para corrigir rapidamente qualquer erro.	should have	M
F3U5	Como médico veterinário, quero que os pontos sejam transparentes e facilitem a precisão, de modo a obter a melhor colocação dos pontos.	nice to have	S
F3U6	Como médico veterinário, quero que a aplicação exiba uma legenda sobre a imagem enquanto marco os pontos, para saber em que zona do cavalo cada ponto deve estar.	must have	S
F4	Como médico veterinário, quero um ecrã com as informações do cavalo.	nice to have	L
F4U1	Como médico veterinário, quero ver num ecrã as seguintes informações do cavalo: quem é o proprietário, quem é o cuidador (se existir), a idade, uma fotografia de perfil, as fotografias das castanhas nos membros do cavalo, os últimos 3 PDFs de hemogramas e a lista de todas as consultas.	nice to have	L
F4U2	Como médico veterinário, quero editar todas as informações disponíveis no cavalo, incluindo quem é o proprietário e o cuidador do cavalo.	nice to have	M
F4U3	Como médico veterinário, quero abrir o perfil do proprietário do cavalo e do seu cuidador e ver os seus contactos e todos os cavalos a eles associados.	nice to have	M
F5	Como médico veterinário, quero ver todas as informações clínicas de uma consulta num cavalo.	nice to have	L

F5U1	Como médico veterinário, quero ver o PDF do hemograma sem sair da aplicação.	nice to have	L
F5U2	Como médico veterinário, quero ver os valor dos batimentos por minuto do coração, o tempo entre ondas de um eletrocardiograma (ECG), os valores afetos à tensão muscular (a frequência em hertz, a rigidez em newton por metro e o tempo de relaxamento em milissegundos) e o nível de claudicação de cada membro.	nice to have	L
F6	Como médico veterinário, quero ter acesso à aplicação através de uma conta pessoal e intransmissível.	must have	L
F6U1	Como médico veterinário, quero poder aceder à aplicação móvel através do <i>login</i> , inserindo os meus dados pessoais, endereço de email e password.	must have	M
F6U2	Como médico veterinário, quero poder alterar a minha password	must have	M
F6U3	Como médico veterinário, quero poder criar uma conta com os meus dados pessoais, endereço de email e password.	must have	M
F7	Como médico veterinário quero obter um raio-x genérico de um membro de um cavalo a partir da fotografia do mesmo.	must have	L
F7U1	Como médico veterinário quero ler uma legenda com o nome de alguns dos ossos presentes no raio-x.	must have	L
F7U2	Como médico veterinário quero ler a legenda apenas quando clico num osso que já está marcado como tendo uma legenda disponível.	must have	L

Tabela 2: Requisitos Funcionais

Requisitos Não Funcionais

ID	Requisito	Escala 'have'
NF1	O médico veterinário, deverá ter credenciais válidas, para conseguir entrar na sua conta pessoal.	must have

NF2	A aplicação deverá estar disponível para telemóveis Android e iOS.	must have
NF3	A aplicação deverá funcionar em qualquer modelo de telemóvel.	must have
NF4	A aplicação deve respeitar a legislação em vigor, assim como as normativas da Comissão Nacional de Proteção de Dados (RGPD).	must have
NF5	No preenchimento de dados online a aplicação deve exigir aos utilizadores a aceitação da política de privacidade.	must have
NF6	A aplicação deverá ter implementada uma cifra de encriptação nas passwords dos utilizadores da mesma.	must have

Tabela 3: Requisitos Não Funcionais

Requisitos Técnicos

ID	Requisito	Escala 'have'
T1	A aplicação necessita de uma conexão à internet estável para poder ser acedida.	must have
T2	A aplicação deverá conseguir aceder e recolher dados da base de dados.	must have
T3	A aplicação deverá estar sempre atualizada.	must have

Tabela 4: Requisitos Técnicos

4.1.2 Descrição detalhada dos requisitos principais

Requisito F1

Como utilizador, quero que depois de fotografar um cavalo possa verificar que a fotografia tem boa qualidade antes de a enviar para processamento.

Após capturar a imagem de um cavalo, o utilizador deverá ter acesso imediato a uma pré-visualização da fotografia, antes de esta ser submetida, podendo escolher se pretende tirar outra fotografia. Desta forma, será possível garantir que a imagem submetida é adequada para ser analisada pelo algoritmo de *machine learning*.

Requisito F2

Como médico veterinário, quero ver o resultado do processamento, o BW e o BCS previstos e a fotografia com os pontos marcados pelo algoritmo.

O médico veterinário deve conseguir visualizar os dados processados pelo sistema e a fotografia deve apresentar claramente as marcações dos pontos identificados pelo algoritmo de *machine learning*. A interface também deverá contemplar os dados de forma clara e organizada, para facilitar a interpretação dos mesmos e validar visualmente os resultados.

Requisito F3

Como médico veterinário, quero editar a posição dos pontos marcados na fotografia e enviar para processamento para o algoritmo de *machine learning*.

O médico veterinário deve conseguir interagir com a fotografia, podendo ajustar os pontos identificados nas fotografias selecionando-os e modificando-os. Neste sentido, a interface deve ser intuitiva, com ferramentas simples de edição, para reposicionar os pontos na imagem. Após ajustes, o médico veterinário deve poder enviar os dados alterados para processamento, para que o sistema recalcule os valores de BW e BCS.

Requisito F4

Como médico veterinário, quero um ecrã com as informações do cavalo.

Para a gestão de informações relacionadas com os cavalos e os seus cuidadores, é necessário disponibilizar um ecrã onde o veterinário possa consultar informações completas sobre cada cavalo. Esse ecrã inclui dados como o nome do proprietário, o cuidador (se houver), a idade do cavalo, uma fotografia de perfil, imagens específicas das castanhas nos membros, os três últimos hemogramas em formato PDF e um histórico completo de todas as consultas realizadas.

A plataforma permitirá ao veterinário editar todas as informações associadas ao cavalo. Esta funcionalidade abrange desde a atualização de dados básicos até à alteração de informações sobre o proprietário ou cuidador, garantindo que os registos permanecem sempre precisos e atualizados.

Por fim, neste ecrã é necessário possibilitar o acesso a um ecrã de perfis do proprietário e do cuidador do cavalo. Este ecrã exhibe todos os cavalos associados a cada pessoa, bem como os respetivos contactos, permitindo uma visão abrangente da relação entre os responsáveis e os seus animais. Estas funcionalidades foram concebidas para otimizar o trabalho clínico e proporcionar um acompanhamento completo e integrado da saúde dos cavalos.

Requisito F5

Como médico veterinário, quero ver todas as informações clínicas de uma consulta num cavalo.

Uma das principais funcionalidades é permitir que o médico veterinário visualize todas as informações clínicas detalhadas de uma consulta relacionada com um cavalo. Esta funcionalidade inclui os dados essenciais recolhidos durante as consultas, oferecendo uma visão geral completa do estado de saúde do animal. Adicionalmente, é possível aceder diretamente ao PDF do hemograma sem ter de sair da aplicação. Embora esta funcionalidade seja considerada um complemento, proporciona uma maior praticabilidade ao médico veterinário, ao evitar interrupções no fluxo de trabalho. Sobre os parâmetros clínicos do cavalo incluímos a frequência cardíaca, o tempo entre ondas de um eletrocardiograma (ECG), as métricas relacionadas com a tensão muscular, como a frequência em hertz, a rigidez em newton por metro e o tempo de relaxamento em milissegundos. Adicionalmente, apresenta o nível de claudicação de cada membro, fornecendo informações para o diagnóstico e o acompanhamento clínico.

De notar que estes requisitos foram cuidadosamente selecionados para atender às necessidades práticas dos veterinários do Hospital Veterinário de Equinos de Santo Estêvão.

Requisito F6

Como médico veterinário, quero ter acesso à aplicação através de uma conta pessoal e intransmissível.

Para o registo de um veterinário na aplicação são necessários o nome, o email, a password, e o nome do hospital a que está ligado, sendo que este último já está previamente registado na base de dados, não sendo por isso, necessário o utilizador preocupar-se com o registo do mesmo.

Requisito F7

Como médico veterinário quero obter um raio-x genérico de um membro de um cavalo a partir da fotografia do mesmo.

Após a 1.^a entrega intercalar, esta funcionalidade surge e é elevada a requisito principal da aplicação. O seu objetivo é proporcionar ao utilizador uma visualização tipo raio-X da zona do cavalo onde a fotografia foi capturada. Esta visualização permite ao veterinário, ou estudante de medicina veterinária, realizar uma análise física preliminar do equídeo diretamente na aplicação, sem a necessidade de recorrer a equipamentos ou aplicações externos.

Segundo o Doutor José Prazeres, esta funcionalidade terá grande utilidade em contextos clínicos e de primeiros contactos com o animal, sendo especialmente relevante como ferramenta de apoio à formação académica e prática de estudantes de medicina veterinária.

4.1.3 Use Cases

Com a utilização de um diagrama de *Use Cases* podemos definir as relações entre os atores e a aplicação em questão. Assim sendo foi desenvolvido um diagrama, na figura 6, que mostra todas as interações que os médicos veterinários terão com o sistema em questão.

4.2 Modelação

Tabela *Clients*

Esta tabela armazena as informações sobre os clientes como os seus identificadores, *name*, *email* e *phoneNumber*. A coluna *isHorseOwner* serve para identificar se o cliente que leva o cavalo a uma consulta veterinária é o proprietário ou o cuidador do animal.

Tabela *Horses*

Contém as informações que permitem identificar os cavalos, o *name*, *profilePicture* e a *birthDate*. Para estudos futuros, incluímos também as fotografias necessárias para a identificação dos cavalos através da castanha nos membros. [San+24]

Tabela *Clients_Horses*

Esta tabela estabelece a associação entre as tabelas *Clients* e *Horses*. É necessária para associar um cavalo ao seu proprietário e ao seu cuidador, bem como para associar

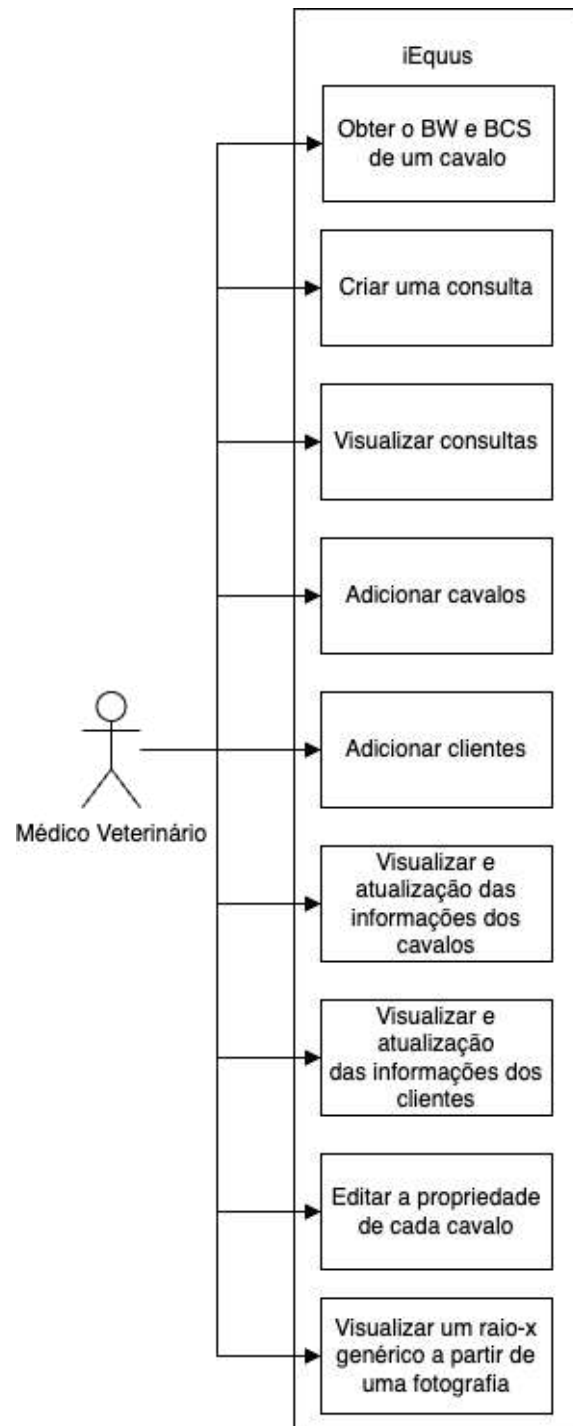


Figura 6: Diagrama de *Use Cases*

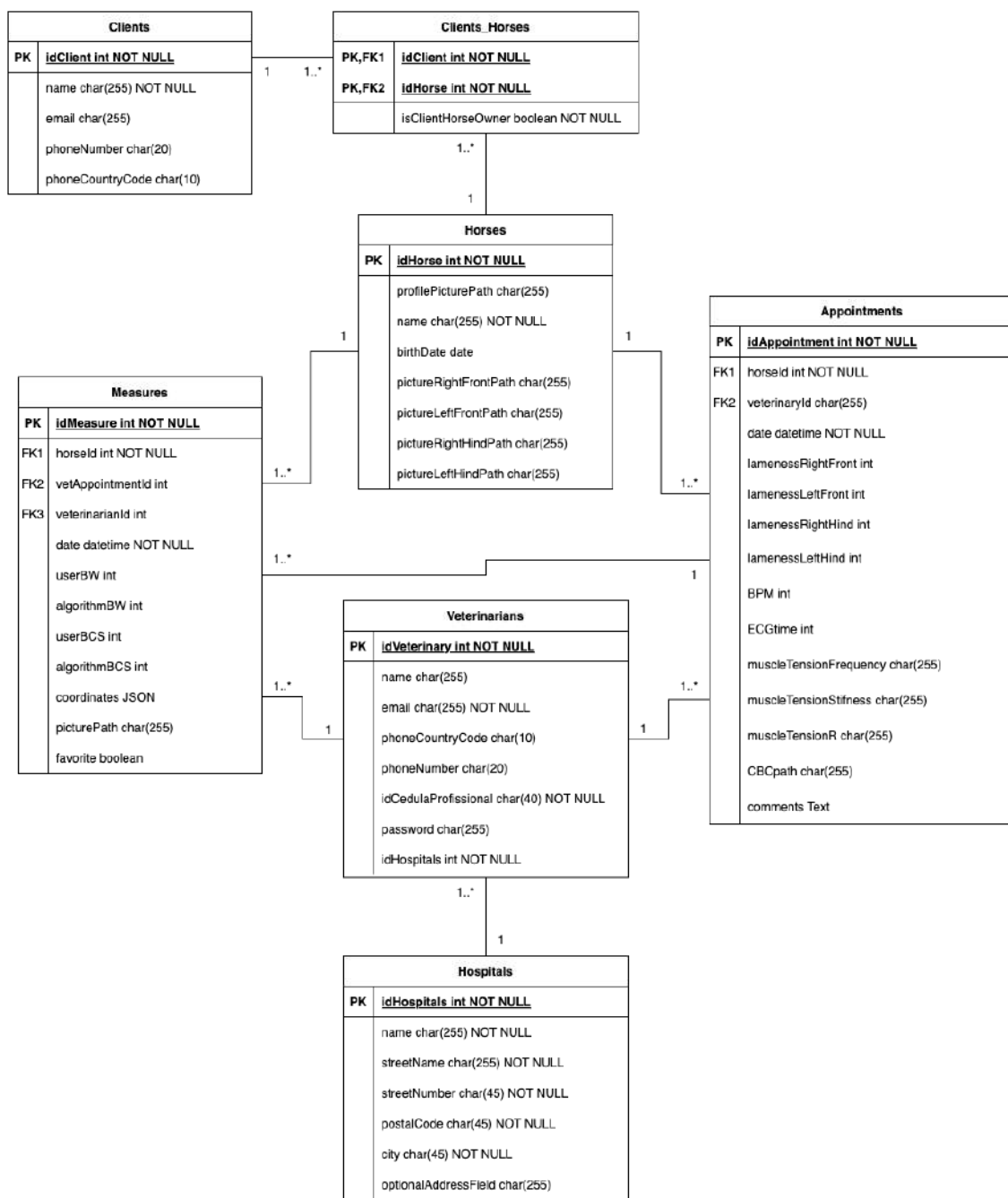


Figura 7: Diagrama entidade-relação na terceira forma normal

os proprietários e cuidadores aos seus animais. A coluna *isClientHorseOwner* permite diferenciar um proprietário de um cuidador.

Tabela CBC

A tabela CBC (*Complete Blood Count*) regista o caminho para os ficheiros PDF de cada hemograma e a data em que as análises foram realizadas. Está associada à tabela *Appointments*, a fim de contextualizar os resultados das consultas em que as amostras foram recolhidas.

Tabela Appointments

A tabela de Consultas contém informações detalhadas sobre cada consulta realizada, incluindo a data, a hora e uma avaliação, numa escala de 0 a 5, do membro que claudica (*lamenessRightFront*, *lamenessLeftFront*, *lamenessRightHind* e *lamenessLeftHind*). A informação de cada consulta é completada com as tabelas *Measures* e *CBC*, e está diretamente relacionada com as tabelas *Veterinarians* e *Horses*, que identificam, respetivamente, o médico responsável pela consulta e o animal atendido.

Tabela Veterinarians

A tabela de Veterinários armazena informações essenciais sobre os profissionais, nomeadamente o nome, os contactos e o número da cédula profissional. Está associada à tabela *Appointments*, permitindo identificar o veterinário designado para cada consulta.

Tabela Hospitals

Esta tabela condensa as informações relativas aos hospitais veterinários. Permite que quando um veterinário se registar, ao associar-se a um hospital, tenha acesso aos cavalos associados e subsequentes clientes.

Tabela Pictures

A tabela de fotografias contém as informações sobre as fotografias capturadas dos cavalos para utilização do algoritmo de *machine learning*, incluindo o caminho do ficheiro e a data em que a imagem foi tirada ou guardada pela aplicação. Está associada à tabela *Appointments* para identificar a que consulta cada fotografia pertence e à tabela *Measures* para a associar a uma previsão do algoritmo.

Tabela Measures

A tabela *Measures* regista as medidas relativas à saúde dos cavalos. Contém o BCS e o BW previstos pelo algoritmo de *machine learning* (*algorithmBCS*, *algorithmBW*) e os introduzidos pelo utilizador (*userBCS*, *userBW*), uma chave estrangeira para a fotografia do algoritmo, o valor dos batimentos por minuto (*BPM*), o tempo entre ondas no eletrocardiograma em milissegundos (*ECGtime*) e os indicadores de tensão muscular em três medidas: a frequência em hertz (*muscleTensionFrequency*), a rigidez em newton por metro (*muscleTensionStiffness*) e o tempo de relaxamento em milissegundos (*muscleTensionR*).

Esta tabela está associada à tabela *Appointments*, mas separada desta, de modo a permitir o registo de várias medições em cada consulta, mantendo assim um histórico completo.

4.3 Protótipos de Interface

Nesta secção, apresentamos protótipos iniciais do projeto, nas figuras 8, 9 e 10, elaborados na aplicação Canva. A escolha do Canva deveu-se em grande parte à sua versatilidade e facilidade de utilização, oferecendo uma interface acessível e intuitiva que permite desenvolver protótipos de forma rápida e com uma curva de aprendizagem bastante reduzida.

Os protótipos apresentados representam uma amostra daqueles que vieram a ser utilizados na solução. Salienta-se, contudo, que estes protótipos foram sujeitos a alterações, uma vez que foram criados com o objetivo de orientar o desenvolvimento do projeto durante a fase de *brainstorming*.

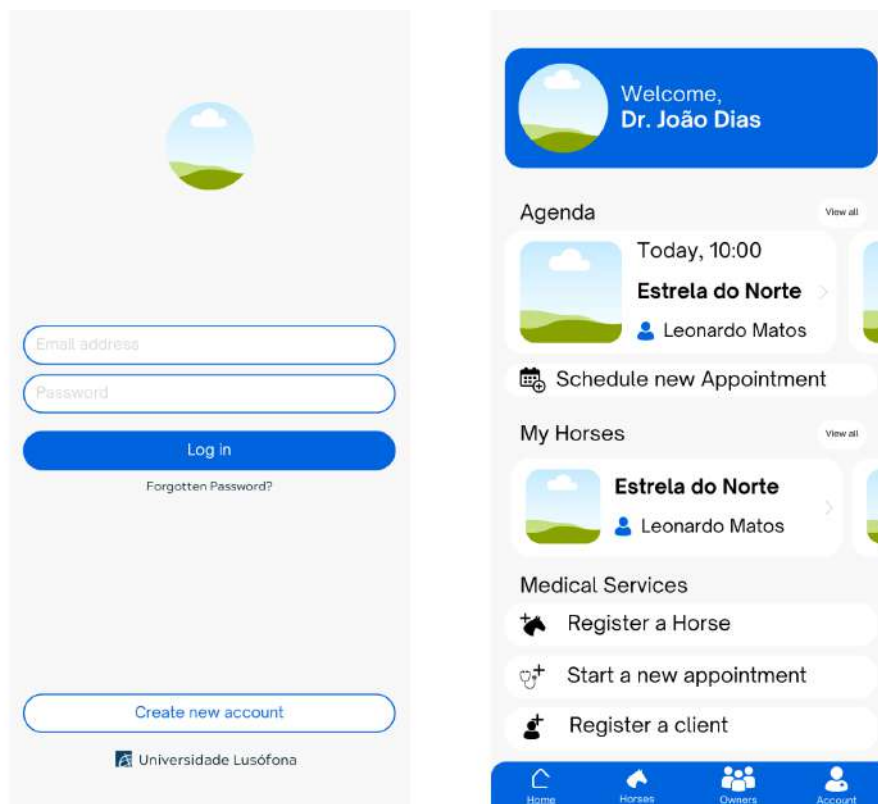


Figura 8: Protótipos dos ecrãs de *Login* e *Home*

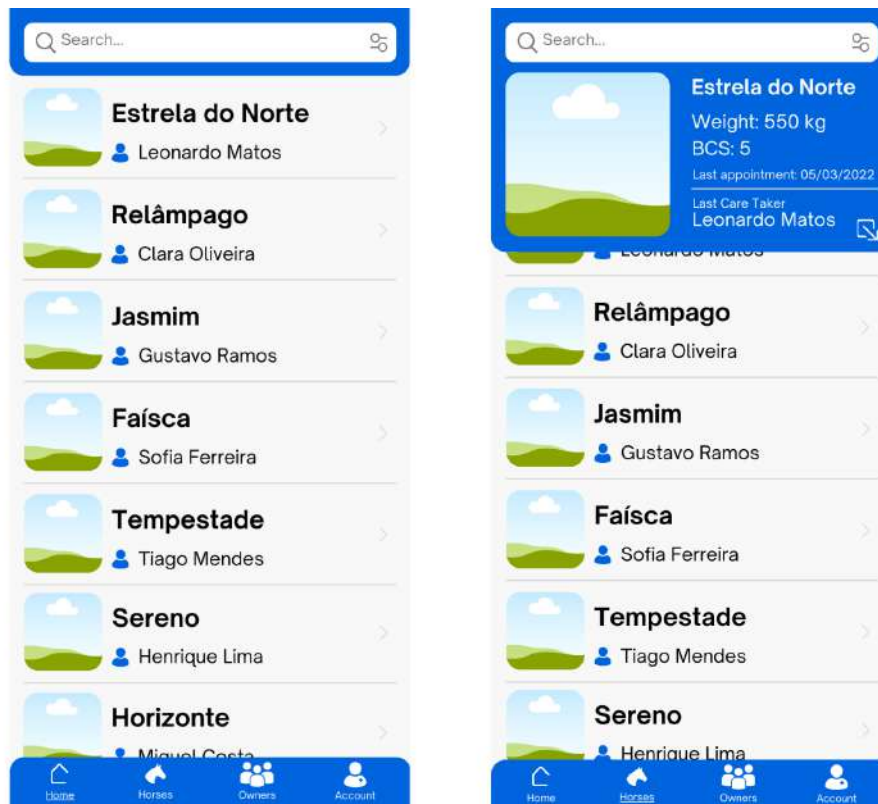


Figura 9: Protótipos dos ecrãs da lista de cavalos

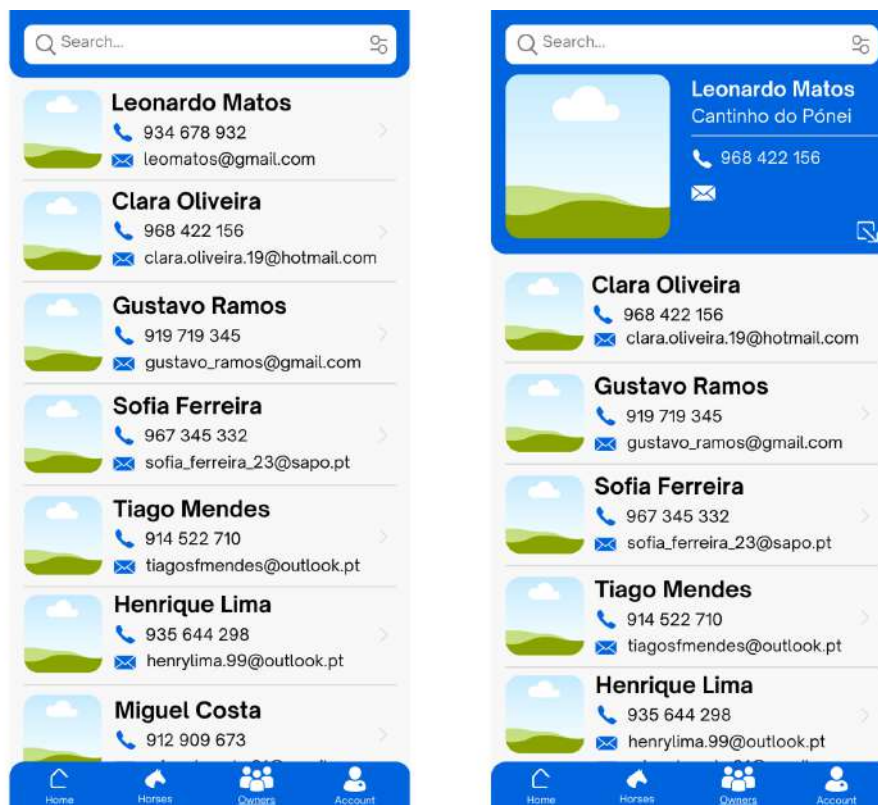


Figura 10: Protótipos dos ecrãs da lista de clientes

5 - Solução

A solução proposta para o iEquus envolve o desenvolvimento de uma aplicação projetada para dispositivos móveis (*smartphones*, Android e iOS), um *back-end* em Flask a funcionar num servidor e uma base de dados MySQL.

A solução terá como principal objetivo permitir que, a partir de uma fotografia de um cavalo, capturada no momento ou previamente armazenada, seja possível obter como *output* uma previsão do BW, um cálculo do BCS e, adicionalmente, um raio-x da zona pretendida do cavalo.

Para este efeito, será necessário que a aplicação transmita tanto a fotografia como a lista de pontos marcados para o servidor, ou apenas a fotografia do membro do cavalo, no caso do raio-x. Neste servidor, estarão armazenados os dados e estará também alojado o algoritmo de *machine learning*. Este algoritmo será responsável pelo processamento das imagens e dos pontos recebidos, gerando como resultado os valores estimados do BW e do BCS e a imagem do raio-x adequado, assim como das legendas e das coordenadas destas.

Como objetivo secundário, pretende-se que a aplicação tenha a capacidade de armazenar um histórico detalhado das informações de saúde de cada cavalo, incluindo dados sobre os seus proprietários e os respetivos cuidadores. Adicionalmente, a aplicação permitirá que os veterinários registem as consultas realizadas, podendo consolidar-se como uma ferramenta abrangente para o acompanhamento da saúde equina.

Para a concretização dos objetivos propostos, será adotada a arquitetura apresentada na Figura 11, cuja explicação é detalhada a seguir.

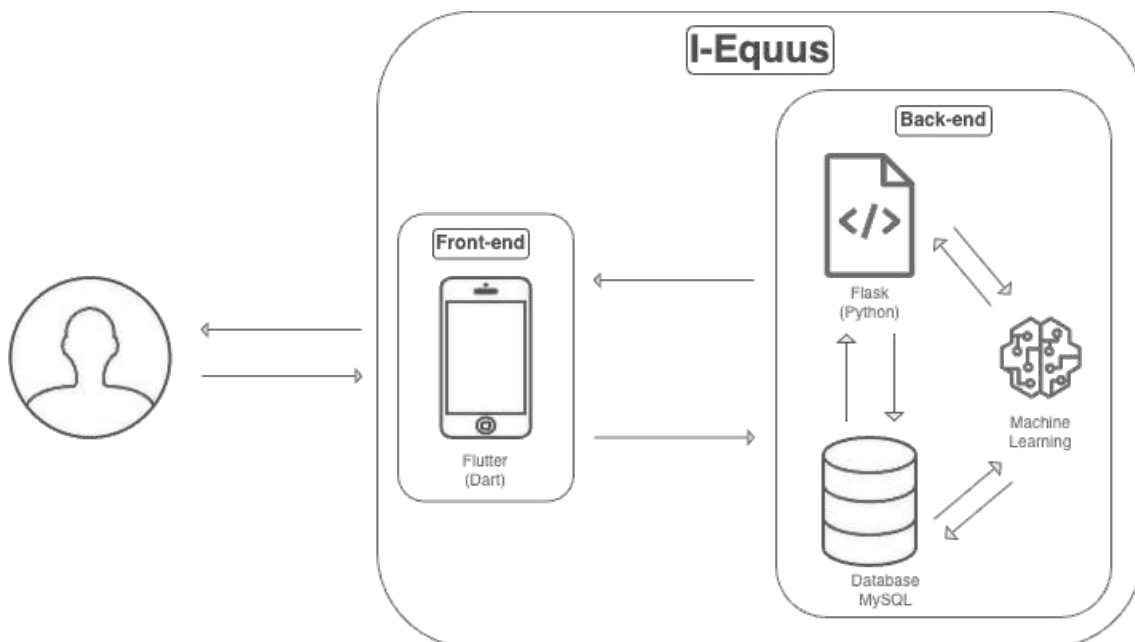


Figura 11: Arquitetura da Solução

5.1 Front-end

O *front-end* é a camada da aplicação responsável pela interface de interação entre o utilizador e o próprio sistema. É através do *front-end* que o utilizador consegue aceder às funcionalidades da aplicação, tornando-se, por conseguinte, o ponto de contacto entre o utilizador e a tecnologia em si. Neste sentido, é importante que seja intuitivo e de fácil navegação, proporcionando uma experiência de utilização eficaz e agradável.

No contexto em concreto do iEquus, o *front-end* está a ser projetado para atender às necessidades dos veterinários e de outros profissionais de saúde animal. A interface permitirá que estes utilizadores façam upload de fotografias dos cavalos para análise do algoritmo e acompanhem os resultados obtidos, como o BCS e o BW. Os veterinários também poderão introduzir mais dados de saúde dos cavalos de modo a manter um histórico mais global dos equídeos, bem como dos seus proprietários e cuidadores. A prioridade foi criar uma interface clara e prática, com foco na facilidade de utilização, permitindo que os profissionais se concentrem na interpretação destes dados, sem complicações na navegação.

5.1.1 Flutter (Dart)

A tecnologia escolhida para o desenvolvimento do *front-end* é o Flutter, uma *framework* que utiliza a linguagem Dart. O Flutter é reconhecido pela sua capacidade de criar aplicações de elevada *performance* para múltiplas plataformas, como iOS, Android, Web, Windows e macOS, a partir de um único código-fonte. O Flutter integra-se facilmente com uma *Application Programming Interface*, facilitando a comunicação com o *back-end*, e é eficiente no desenvolvimento de aplicações com elevado desempenho.

Deste modo, o desenvolvimento do *front-end* em Flutter terá como objetivo proporcionar uma experiência de utilizador eficiente, desde o envio das imagens dos equinos até à visualização dos resultados. Espera-se que a interface permita o envio rápido de fotografias para a análise e a apresentação dos resultados de forma clara.

5.2 Back-end

O *back-end* da aplicação é a camada do sistema responsável pelo processamento e pela gestão dos dados, bem como pela implementação da lógica de negócio. Esta parte do sistema gere todas as operações que ocorrem nos "bastidores" da aplicação, nomeadamente a interação com a base de dados e com o algoritmo de *machine learning*. O desempenho do *back-end* permite que o *front-end* tenha acesso às informações e funcionalidades essenciais, realizando um trabalho de processamento e armazenamento necessários para que a aplicação funcione de forma eficiente e precisa.

No caso do iEquus, o *back-end* é responsável pela receção e pelo processamento das imagens submetidas pelos utilizadores. Este processamento envolve o processamento das coordenadas nas fotografias dos cavalos, a determinação do BCS e do BW, ou do raio-x correto, e o envio dos resultados para o *front-end*.

5.2.1 Flask (Python)

O Flask é uma *framework* em Python, em que as suas principais vantagens são a simplicidade e a flexibilidade, que permitem a criação de aplicações rápidas e eficientes. No caso do iEquus, o Flask atua como intermediário entre o *front-end* e os componentes internos do sistema, que recebe os pedidos do utilizador, processa-os e comunica-os à base de dados e ao modelo de *machine learning*.

5.2.2 Algoritmo de *Machine Learning*

O algoritmo de *machine learning* que será utilizado para o iEquus fundamenta-se no estudo que serve de base a este projeto, de Urbanek e Zebeli. [UZ23] Concretamente, o algoritmo foi treinado para calcular de forma precisa o BCS e o BW dos equinos a partir das imagens fornecidas e é capaz de realizar previsões sobre o estado de saúde dos animais. À medida que o sistema é alimentado com novos dados, o modelo será continuamente ajustado e aperfeiçoado, aumentando a sua precisão ao longo do tempo. Este processo de melhoria contínua é fundamental para garantir que os resultados oferecidos são cada vez mais fiáveis e permanecem úteis para os utilizadores.

É importante salientar que esta componente de *machine learning* não se insere no âmbito deste trabalho, sendo fornecida e integrada de forma adequada através da disponibilização dos ficheiros de modelos de inteligência artificial por parte dos orientadores deste TFC. Para tal, foram disponibilizados dois ficheiros:

- scaler.pkl - garante que os dados introduzidos mais tarde no sistema são transformados da mesma forma que foram durante o treino do modelo;
- model.pkl - contém o modelo de machine learning propriamente dito, isto é, o algoritmo que foi treinado com os dados (já transformados pelo scaler) para prever o BW e o BCS dos cavalos.

Para além destes, com a nova funcionalidade de raio-X de zonas do cavalo a partir de uma fotografia, surgiu a necessidade de adicionar um novo algoritmo de inteligência artificial, que reconheça a parte corporal do cavalo pretendida e devolva uma imagem, genérica e pré-processada, de um raio-x com as legendas dos ossos e as suas respetivas coordenadas. Deste modo, a aplicação em Flutter poderá receber estas coordenadas e mapeá-las na imagem, para que o utilizador interaja com a mesma e as suas legendas.

5.2.3 MySQL (Base de Dados)

A base de dados MYSQL é o sistema de gestão de bases de dados relacional que irá armazenar as informações inerentes à aplicação iEquus. Esta ferramenta permite organizar e gerir eficientemente grandes volumes de dados, garantindo a integridade e a acessibilidade da informação.

A base de dados será responsável pelo armazenamento do caminho no servidor das imagens enviadas pelos utilizadores, bem como dos dados dos cavalos e dos resultados das análises realizadas no âmbito das funcionalidades da aplicação. Para além disto, a base de dados permitirá o registo e o acompanhamento do histórico das interações, facilitando a análise global dos dados e o fornecimento de informações de fácil acesso aos utilizadores.

A base de dados representada na figura 7 é um sistema de gestão de informações relacionadas com cavalos, incluindo proprietários, consultas veterinárias, métricas específicas de saúde de equinos e outros dados associados.

O objetivo global do *back-end* do iEquus é garantir que o processamento, o armazenamento e a análise dos dados sejam realizados de forma integrada, precisa e segura. O Flask será responsável pela comunicação entre o *front-end*, enquanto o MySQL garantirá que os dados estão organizados e acessíveis ao servidor em Flask para poder disponibilizá-los corretamente. À medida que o sistema evolui, o algoritmo de *machine learning* fornecerá métricas de saúde cada vez mais precisas. Com estas tecnologias, espera-se que o *back-end* seja capaz de oferecer resultados em tempo real, otimizando a interação dos utilizadores com a aplicação e fornecendo dados fundamentais para o acompanhamento e melhoria da saúde dos cavalos.

5.3 Arquitetura da Solução

A arquitetura da solução iEquus foi delineada para viabilizar uma comunicação eficiente entre o *front-end* e o *back-end* do sistema, utilizando uma APIs que possibilita a troca de dados e a execução de operações em tempo real. Ao separar claramente as camadas de apresentação e de processamento, este modelo de arquitetura não só contribui para uma gestão mais eficaz do sistema, como também garante a sua escalabilidade, facilitando a implementação de melhorias futuras. A separação entre as duas partes permite a adaptação do sistema a novos requisitos sem comprometer o seu desempenho e flexibilidade.

Em suma, a arquitetura proposta para o iEquus proporciona uma estrutura robusta e flexível, que não só assegura a eficiência no processamento e transmissão de dados entre as diversas camadas do sistema, como também oferece a escalabilidade necessária para futuras adaptações e melhorias. A separação entre as camadas de apresentação e processamento permite a adaptação e evolução contínuas do sistema, assegurando que a solução permanece eficaz ao longo do tempo.

5.4 Solução desenvolvida

A solução desenvolvida para o projeto *iEquus* foi organizada em três repositórios principais, disponibilizados no GitHub:

- *Front-end*: github.com/DEISI-ULHT-TFC-2024-25/TFC-DEISI2006-iEquus-Front-end.
- *Back-end*: github.com/DEISI-ULHT-TFC-2024-25/TFC-DEISI2006-iEquus-Back-end.
- Servidor de Previsão com os Modelos de *Machine Learning*): github.com/DEISI-ULHT-TFC-2024-25/TFC-DEISI2006-iEquus-Predict.

A apresentação deste TFC fica neste vídeo no Youtube:

- <https://youtu.be/f7LeDrbPwFc>

6 - Calendário

Para suporte ao desenvolvimento do TFC e organizar as tarefas inerentes, foi elaborado um Cronograma de Gantt na figura 12, que define o desenho temporal do projeto. Este diagrama proporciona uma visão de alto nível do progresso das atividades e serviu como uma ferramenta de apoio à gestão, facilitando o cumprimento dos prazos estabelecidos para as entregas e o planeamento estruturado das diferentes tarefas. O cronograma está dividido em três fases principais, cada uma correspondente a uma das entregas planeadas e com as suas respetivas datas de entrega. Estas fases refletiram os diferentes *outputs* esperados e funcionaram como marcos para avaliar o progresso do projeto em momentos específicos.

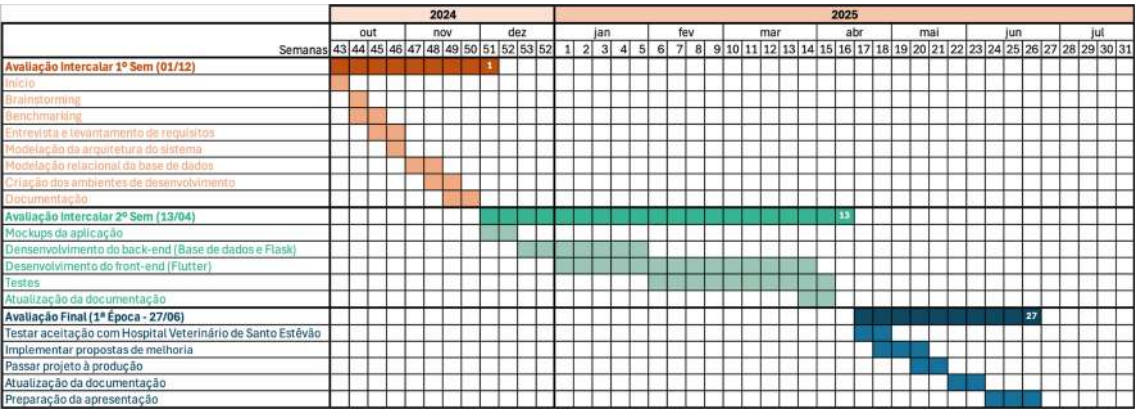


Figura 12: Cronograma, de alto nível, das atividades do projeto

Na 2ª fase do projeto, os entregáveis previstos fase foram concluídos, e iniciou-se o trabalho relativo aos entregáveis da fase final. A sequência de desenvolvimento de tarefas, tal como definida no planeamento, considerou as dependências entre atividades. Por exemplo, a modelação da arquitetura do sistema dependeu da realização da entrevista e do levantamento de requisitos. De forma semelhante, o desenvolvimento do *front-end* foi condicionado pelo desenvolvimento dos protótipos, enquanto que os testes só puderam ser realizados a partir de algum desenvolvimento do *back-end* e do *front-end*.

Com o objetivo de otimizar a integração das componentes do sistema, numa fase inicial, foi decidido priorizar o início do desenvolvimento do *back-end* antes do *front-end*. Esta abordagem permitiu assegurar que a comunicação entre o servidor e a aplicação estavam implementados antes de se avançar para a interface com o utilizador. Assim, na 2ª fase do projeto, o *back-end* estava totalmente desenvolvido e em fase de implementação num servidor da universiade Lusófona. Já o *front-end* em fase de desenvolvimento, uma vez que sofreu ajustes nos requisitos a meio do processo de implementação.

Este planeamento procurou estabelecer um fluxo de trabalho claro e eficiente, de modo a mitigar riscos e otimizar os recursos disponíveis, sendo este um trabalho individual. O foco em dependências entre tarefas e validações progressivas contribuiu para assegurar que os objetivos definidos no início do projeto foram efetivamente alcançados ao longo de todas as fases de desenvolvimento.

7 - Plano de Testes e Validação

Sendo que a aplicação se destina à utilização pela Clínica de Santo Estêvão e por alunos da Faculdade de Medicina Veterinária da Universidade Lusófona, foram planeados alguns testes acompanhados por um questionário, no anexo 9.1, e por um manual de instalação da aplicação para Android, no anexo 9.2, com o objetivo de avaliar esta versão ainda embrionária do iEquus.

Contudo, devido a um atraso na passagem do *back-end* para produção, esses testes ainda não puderam ser realizados pelos referidos utilizadores. Deste modo, a sua execução decorrerá enquanto o *back-end* se encontrar alojado na solução temporária de um servidor doméstico, conforme descrito na secção 8.2, até que seja dada a continuidade à tarefa de alojamento do *back-end* deste TFC nos servidores da Universidade Lusófona para o efeito.

Título	Descrição	Requisitos	Resultado esperado
Registo com uma conta pessoal	O utilizador preenche o formulário de registo de utilizador.	F6, F6U3	O utilizador preenche o formulário de criação de uma conta com os respetivos dados e obtém uma mensagem de sucesso.
Login com uma conta pessoal	O utilizador preenche o formulário de login de utilizador.	F6, F6U1	O utilizador preenche o formulário de <i>login</i> e consegue aceder à aplicação com a sua conta pessoal.
Obtenção de uma previsão de BW e BCS	O utilizador obtém uma previsão do BW e do BCS adequados	F1, F1U1, F1U2, F1U4, F1U5, F1U6, F2, F2U1, F2U2, F2U3, F3, F3U1, F3U3, F3U4, F3U6	O utilizador a partir de uma fotografia, marca os pontos necessários, revê-os e obtém uma previsão do BW e do BCS adequados.

Visualização do perfil de um cavalo	O utilizador vê as informações de um cavalo através do ecrã de perfil do mesmo. Visualiza também as medições de BW e BCS do mesmo cavalo.	F2. F2U4, F4, F4U1	O utilizador vê o ecrã de perfil de um cavalo, os seus proprietários e/ou cuidadores, caso existam, e o histórico de resultados das medições de BW e BCS provenientes do algoritmo de <i>machine learning</i> e da introdução das correções por parte dos utilizadores dos mesmo dados.
-------------------------------------	---	--------------------	---

Tabela 5: Tabela de Testes

8 - Resultados

8.1 *Update*: 27 de abril de 2025

O desenvolvimento do projeto apresentou-se com alguns avanços comparativamente à 1ª entrega intercalar. No entanto, depois desta entrega e depois de reunião com o docente da faculdade de Medicina Veterinária da Universidade Lusófona, o Doutor José Prazeres, ficaram definidas algumas alterações aos requisitos F4 e F5 e adicionou-se o requisito F7. Os primeiros passam a ser qualificados na escala 'have' como 'nice to have', para dar lugar a este requisito funcional, o F7, que passa a ser prioritário, já presente nas tabelas 2 e 7.

8.1.1 Progresso do trabalho dos requisitos

Para avaliar o progresso do trabalho relativamente aos requisitos funcionais, foram acrescentadas duas novas colunas de estado à Tabela 2: uma para indicar os requisitos que já se encontravam concluídos, para tal utilizou-se um 'x', e deixou-se em branco os que ainda não estavam finalizados; e outra coluna destinada a notas adicionais, permitindo incluir observações relevantes sobre requisitos específicos, sempre que necessário.

Requisitos Funcionais

ID	Escala 'have'	Esforço	Estado	Notas de desenvolvimento
F1	must have	L	x	Sub-requisitos 'must have' concluídos. Fotografias podem ser tiradas no momento ou escolhidas da galeria usando <i>ImagePicker</i> .
F1U1	must have	S	x	
F1U2	must have	S	x	
F1U3	nice to have	S		Requer nova implementação. A <i>ImagePicker</i> não oferece suporte nativo a sobreposições com molduras para alinhamento do cavalo.
F1U4	should have	S		Não existe tutorial específico, mas há orientações visuais implícitas após a tiragem dos pontos que ajudam a guiar o utilizador.
F1U5	must have	S	x	
F1U6	must have	S	x	

F2	must have	L	x	
F2U1	must have	M	x	
F2U2	must have	S	x	
F2U3	must have	S	x	
F2U4	must have	L		
F2U5	should have	S		
F3	must have	L	x	
F3U1	nice to have	L	x	
F3U2	nice to have	M		
F3U3	must have	S	x	
F3U4	should have	M		Neste momento, apenas a funcionalidade <i>undo</i> está a funcionar, sendo que, para o <i>redo</i> da marcação de um ponto, tem de se voltar a colocá-lo.
F3U5	nice to have	S		
F3U6	must have	S	x	
F4	nice to have	L	x	
F4U1	nice to have	L	x	Para este requisito, falta desenvolver as funcionalidades relativas a uma consulta, isto é, a lista de consultas de cada cavalo e os ficheiros PDF's dos últimos 3 hemogramas, cujo desenvolvimento passou para depois da nova funcionalidade de raio-x.
F4U2	nice to have	M	x	
F4U3	nice to have	M		
F5	nice to have	L		
F5U1	nice to have	L		
F5U2	nice to have	L		
F6	must have	L	x	

F6U1	must have	M	x	
F6U2	must have	M		
F6U3	must have	M		
F7	must have	L	x	
F7U1	must have	L		
F7U2	must have	L		

Tabela 6: Resultados: Requisitos Funcionais

F1

Para este requisito, os sub-requisitos classificados como 'must have' encontram-se concluídos, ficando apenas por implementar alguns dos 'nice to have'.

A solução já permite a utilização de fotografias tiradas no momento ou provenientes da galeria, na figura 14b, e a utilização do flash do telemóvel, na figura 14c, recorrendo à classe *ImagePicker* do Flutter, que fornece estas funcionalidades através dos métodos nativos do Android e iOS. Desta forma, para implementar o requisito F1U3, seria necessário desenvolver uma nova solução, pois a classe *ImagePicker* não suporta a sobreposição de molduras que facilitem o alinhamento do cavalo no enquadramento.

Relativamente ao F1U4, não existe um tutorial específico e explícito que explique como deve ser tirada a fotografia, no entanto, existem já indicações visuais no ecrã antes de se tirar a fotografia que orientam o utilizador quanto ao resultado esperado após a marcação dos pontos, na figura 14a, o que, por sua vez, fornece pistas sobre como a fotografia deve ser capturada. No entanto, ao longo do processo de marcação das linhas há uma legenda e a possibilidade de se abrir um *popup*, como mostra a figura 14f que ajude e identifique no cavalo a localização morfológica para a linha a marcar.

F2

Este requisito é fundamental para este TFC, dado que é um requisito central. Está quase terminado, como se pode perceber nas figuras 15, faltando apenas a visualização de um histórico dos resultados anteriores para concluir os requisitos classificados como 'must have' na escala 'have' definida anteriormente. Este foi, sem dúvida, o requisito que mais tempo ocupou de todo o TFC até à data. Isto porque, é necessário utilizar as coordenadas da imagem em relação com as coordenadas do ecrã do dispositivo móvel e é necessário pintar a imagem em tempo real, isto é, à medida que o utilizador marca os pontos, traços estes que trouxeram algumas dificuldades para o desenvolvimento.

No que toca ao *back-end*, este requisito obriga também a que o *endpoint* do servidor Flask tenha várias interações, desde receber a imagem com as coordenadas, retornar os valores de BW e BCS e, ainda, algumas informações sobre a medição criada e guardada na base de dados (como o identificador único).

De notar que, não estando ainda o algoritmo de *machine learning*, desenvolvido pelos orientadores deste TFC, preparado para ser utilizado neste trabalho, os valores retornados pelo *back-end* para o *front-end* ainda não são reais, mas sim aleatórios.

F3

O requisito F3, nas figuras 14, permite editar os pontos na imagem durante o processo de marcação. Neste momento, já é possível apagar os pontos, um de cada vez. No entanto, não é possível refazer uma posição anterior, sendo que, para isso, é necessário recolocar manualmente os pontos. A funcionalidade de *undo* foi desenvolvida para apagar apenas as coordenadas visíveis no momento e como as coordenadas são colocadas em pares e há um ecrã para cada par, tendo em conta a linha que se quer marcar no cavalo, a ferramenta de *undo* apaga apenas os pontos que formam essa linha presente em cada ecrã. Para navegar entre os ecrãs das linhas há botões que o permitem, sendo que navegação para a linha seguinte está vedada apenas para quando a linha atual está formada, como se pode observar na figura 14e.

Para este requisito, já está implementada a legenda e as cores das linhas que foram atribuídas de forma específica para cada uma, com o objetivo de orientar o utilizador sobre qual a linha deve marcar em cada momento. Existem 7 linhas no total, o que facilita a identificação de cada uma delas.

F4

Para o ecrã onde estão concentradas as informações de um cavalo, na figura 13e, já está implementada a identificação do proprietário, do cuidador(se existir), a idade (apenas pela data de nascimento), a fotografia de perfil e, apesar de no momento de criação de um novo cavalo já estarem a ser guardadas as fotografias das castanhas dos membros, ainda não estão a ser disponibilizadas neste ecrã. Fica a faltar desenvolver as funcionalidades relativas a uma consulta, isto é, a lista de consultas de cada cavalo e os ficheiros PDF's dos últimos 3 hemogramas, cujo desenvolvimento passou para depois da nova funcionalidade de raio-x.

F5

Este requisito a partir da mais recente reunião com o Doutor José Prazeres no dia 24 de março de 2025 passou a 'nice to have', isto é, o ecrã de consultas, nas figuras 17, que deverá congrega as funcionalidades de visualização e introdução do hemograma, do valor dos batimentos por minuto do coração, do tempo entre ondas de um eletrocardiograma (ECG), dos valores afetos à tensão muscular (a frequência em hertz, a rigidez em newton por metro e o tempo de relaxamento em milissegundos) e do valor do nível de claudicação de cada membro. No entanto, a base de dados e a API em Flask já estão preparadas para receber, processar e armazenar estas informações. Deste modo, da parte do *back-end* está finalizado, ficando apenas a faltar implementar no *front-end*.

F6

A funcionalidade de *login* no *front-end*, presente na figura 13a, está implementada restando apenas a implementação da recuperação de password e de um ecrã para registo de um novo utilizador. No entanto, no *back-end*, todas as funcionalidades relacionadas com o *login* já estão operacionais, incluindo o *login* em si, a alteração da password e o registo de novos utilizadores.

O sistema de autenticação utiliza um *JSON Web Token* para cada *login* para garantir sessões seguras e sem necessidade de manutenção de estado de cada cliente no servidor. Após o utilizador fornecer umas credenciais válidas, o servidor gera um *token* JWT assinado, que é enviado ao *front-end* e armazenado localmente. Este *token* é incluído automaticamente em cada pedido subsequente como cabeçalho de autorização, permitindo a validação da identidade do utilizador em cada requisição sem necessidade de

uma reautenticação constante. De notar que, para a correta utilização da chave secreta do servidor Flask na geração dos JWTs, está devidamente guardada numa variável de ambiente, assim como as credenciais de acesso ao servidor da base de dados MySQL, garantindo maior segurança.

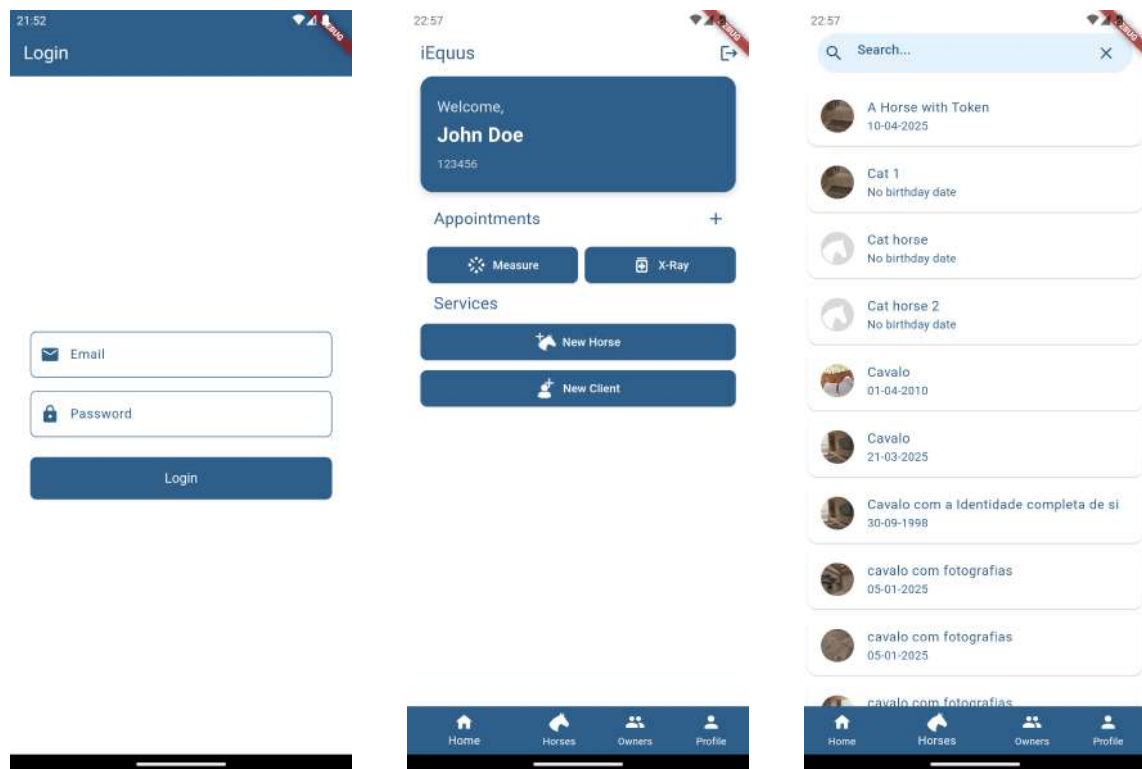
F7

O objetivo deste novo requisito, representado nas figuras 16, é permitir que, a partir de uma fotografia tirada aos membros do cavalo, o *back-end* receba essa imagem, comunique com a API desenvolvida pelos orientadores deste TFC, e obtenha uma imagem de raio-x com legendas associadas, mapeadas através de coordenadas. Posteriormente, essa informação é devolvida à aplicação móvel, que deverá exibir a imagem de raio-x e mapear as legendas recebidas nas posições correspondentes, permitindo que, ao tocar na imagem, o utilizador visualize a respetiva legenda.

Para concretizar este requisito, foi necessário implementar os *endpoints* correspondentes na API e desenvolver o ecrã específico no *front-end*.

Nesta entrega, já é possível visualizar o ecrã desenvolvido, onde a aplicação Flutter envia uma imagem da perna do cavalo para o servidor Flask e recebe de volta uma imagem de raio-x. No entanto, ainda está por implementar o mapeamento interativo das legendas na imagem. Para além disto, a parte da API responsável por fornecer os dados das legendas está a ser desenvolvida pelos orientadores deste TFC; até ao momento, o *back-end* apenas devolve uma imagem de raio-x com legendas genéricas.

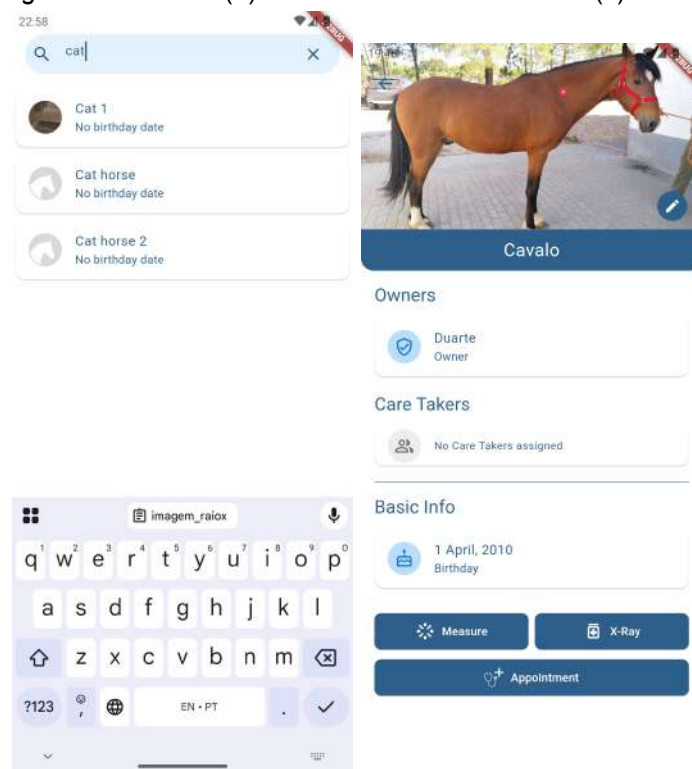
8.1.2 Screenshots do *front-end*



(a) Ecrã de *login*

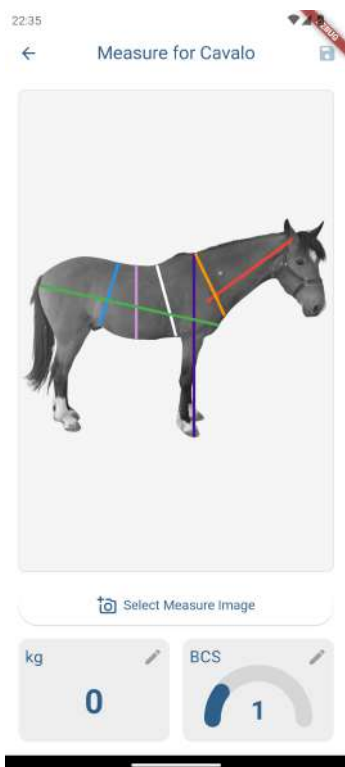
(b) Ecrã de *dashboard*

(c) Ecrã da lista de cavalos

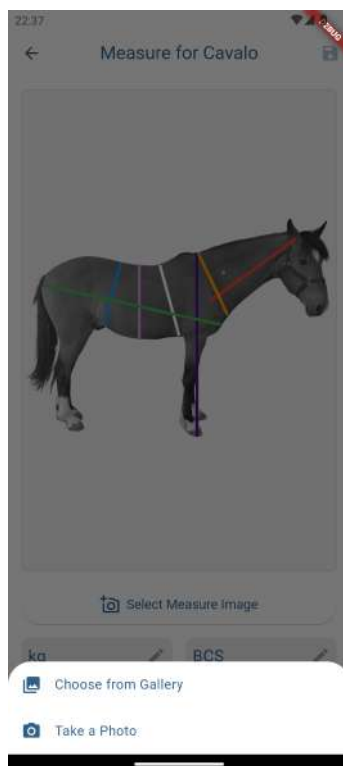


(d) Ecrã da lista de cavalos com a funcionalidade de pesquisa
(e) Ecrã de perfil de um cavalo

Figura 13: Screenshots dos ecrãs de *login*, da *dashboard*, da lista e perfil de cavalos



(a) Ecrã para a criação de uma medição do BW e BCS vazio



(b) Menu para a escolha da origem da fotografia



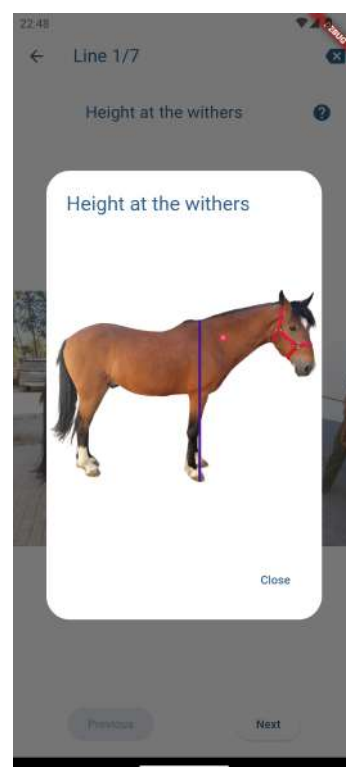
(c) O *Image Picker* nativo do Android, câmara



(d) O *Image Picker* nativo do Android, confirmação da fotografia



(e) Ecrã para a marcação das coordenadas

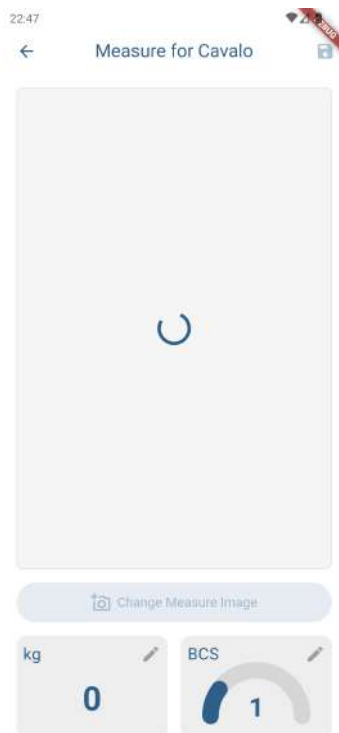


(f) *Popup* para ajuda a marcação das coordenadas

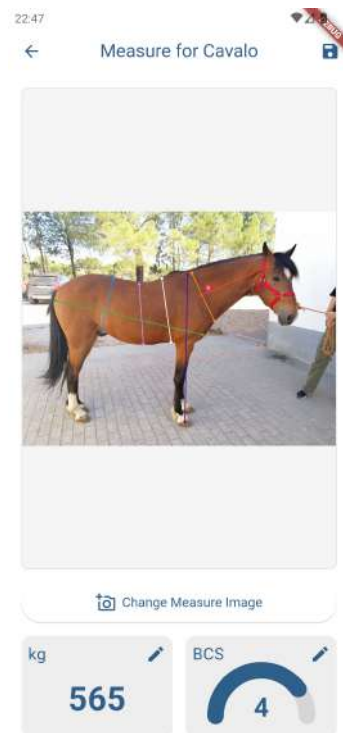
Figura 14: *Screenshots* dos ecrãs para a criação de uma medição de BW e BCS



(a) Ecrã para validar as coordenadas e a marcação das linhas na imagem



(b) Ecrã com o processamento do BW e BCS



(c) Ecrã com o resultado o BW e BCS

Figura 15: *Screenshots* dos ecrãs com o resultado da marcação das coordenadas e o BW e BCS previstos pela API

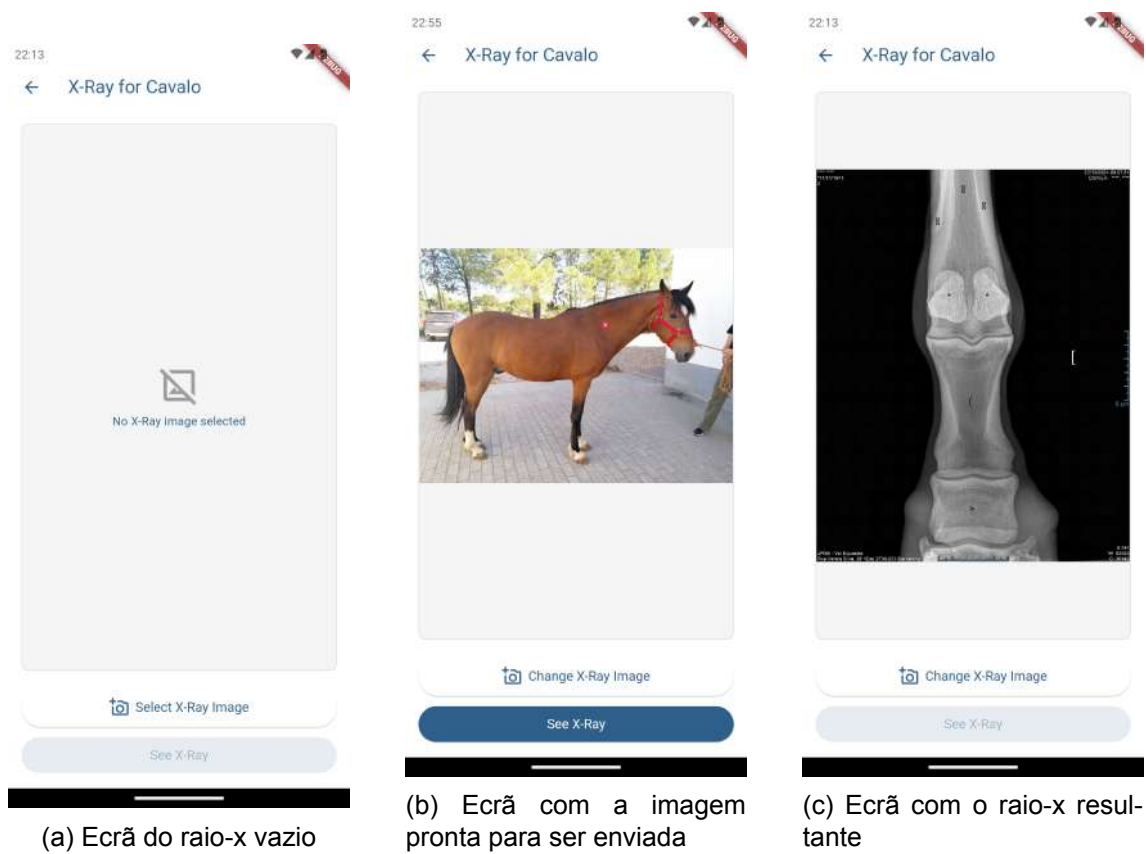


Figura 16: Screenshots dos ecrãs do raio-x



(a) Ecrã de consulta para as informações principais

(b) Ecrã de consulta para criar uma nova medição de BW e BCS dentro de uma consulta

(c) Ecrã de consulta para o médico veterinário deixar alguns comentários acerca da consulta

Figura 17: *Screenshots* do ecrã de consultas

8.2 *Update: 27 de junho de 2025*

Para esta entrega final, o projeto registou avanços significativos, sobretudo na passagem para ambiente de produção, os quais foram apresentados e validados em reunião com o Doutor José Prazeres. No entanto, visto ter existido algumas complicações na comunicação com o professor responsável por gerir o servidor de TFC's da Universidade Lusófona, comunicações estas desde de 3 de abril, e reforçadas nos meses posteriores, decidiu-se que, temporariamente, o *back-end* passaria a produção num servidor doméstico. Este servidor tem como sistema operativo o ubuntu server, sendo que a base para o alojar são três Docker Containers:

- o servidor em Flask que recebe os pedidos do *front-end*;
- o Predict - um servidor em Flask que recebe os pedidos do BW e BCS do servidor anterior e que os retorna;
- o serviço de base de dados em MySQL.

Para garantir os requisitos de segurança, tanto na comunicação entre o *back-end* e o *front-end*, como na proteção do próprio servidor, e ainda para assegurar o encaminhamento correto dos pedidos para o container correspondente, é utilizado um *reverse proxy*, nomeadamente o Nginx. O Nginx atua como intermediário entre os clientes e os serviços da aplicação, permitindo controlar e filtrar o tráfego de entrada. Esta camada adicional permite aplicar políticas de segurança, como a restrição de origens, a limitação

de pedidos, a proteção contra ataques de força bruta e sobretudo a encriptação das comunicações com HTTPS e a gestão de certificados SSL. Neste caso, gerados a partir da autoridade de certificação externa Let's Encrypt, tratando-se de uma autoridade sem fins-lucrativos, pelo que apenas é necessário demonstrar o controlo sobre o domínio. [Enc] Este pertence a um domínio pessoal através do serviço Amen.pt [ame].

Em termos do desenvolvimento do *back-end*, foram implementadas melhorias significativas, nomeadamente a separação da lógica do servidor *iEquus* da componente *Predict*. Esta separação contribui para uma melhor escalabilidade, permitindo que cada serviço possa evoluir de forma independente, e facilita igualmente a manutenção, ao isolar responsabilidades distintas. Adicionalmente, o código foi reorganizado e preparado para ser executado em ambiente de produção num servidor físico, garantindo maior estabilidade e segurança.

Do lado do *front-end*, procedeu-se à finalização do requisito F7, que surgiu a meio do processo de desenvolvimento, à continuação do trabalho no requisito F2, bem como à correção de erros identificados na aplicação, com o objetivo de melhorar a experiência do utilizador e garantir o cumprimento dos requisitos funcionais definidos.

8.2.1 Progresso de trabalho dos requisitos

Do mesmo modo que no 8.1, para avaliar o progresso do trabalho relativamente aos requisitos funcionais, foram acrescentadas duas novas colunas de estado à Tabela 2: uma para indicar os requisitos que já se encontram concluídos, para tal utilizou-se um 'x', e deixou-se em branco os que ainda não estão finalizados; e outra coluna destinada a notas adicionais, permitindo incluir observações relevantes sobre requisitos específicos, sempre que necessário.

Requisitos Funcionais

ID	Escala 'have'	Esforço	Estado	Notas de desenvolvimento
F1	must have	L	x	Sub-requisitos 'must have' concluídos. Fotografias podem ser tiradas no momento ou escolhidas da galeria usando <i>ImagePicker</i> .
F1U1	must have	S	x	
F1U2	must have	S	x	
F1U3	nice to have	S		Requer nova implementação. A <i>ImagePicker</i> não oferece suporte nativo a sobreposições com molduras para alinhamento do cavalo.
F1U4	should have	S	x	Através de orientações visuais explícitas antes, durante e após a tiragem dos pontos que ajudam a guiar o utilizador onde os deve marcar.
F1U5	must have	S	x	

F1U6	must have	S	x	
F2	must have	L	x	
F2U1	must have	M	x	
F2U2	must have	S	x	
F2U3	must have	S	x	
F2U4	must have	L	x	
F2U5	should have	S		
F3	must have	L	x	
F3U1	nice to have	L	x	
F3U2	nice to have	M		
F3U3	must have	S	x	
F3U4	should have	M		Apenas a funcionalidade <i>undo</i> está a funcionar, sendo que, para o <i>redo</i> da marcação de um ponto, tem de se voltar a colocá-lo.
F3U5	nice to have	S		
F3U6	must have	S	x	
F4	nice to have	L	x	
F4U1	nice to have	L	x	Para este requisito, ficou a faltar desenvolver as funcionalidades relativas a uma consulta, isto é, a lista de consultas de cada cavalo e os ficheiros PDF's dos últimos 3 hemogramas, cujo desenvolvimento passou para depois da nova funcionalidade de raio-x.
F4U2	nice to have	M		
F4U3	nice to have	M		
F5	nice to have	L		
F5U1	nice to have	L		
F5U2	nice to have	L		
F6	must have	L	x	

F6U1	must have	M	x	
F6U2	must have	M	x	
F6U3	must have	M	x	
F7	must have	L	x	
F7U1	must have	L	x	
F7U2	must have	L	x	

Tabela 7: Resultados: Requisitos Funcionais

F2

Este requisito, sendo fundamental para este TFC, fica concluído no que respeita aos requisitos classificados como 'must have' na escala 'have' anteriormente definida. Isto deve-se à implementação do requisito F2U4, ilustrado na Figura 18b, bem como à integração do algoritmo de inteligência artificial fornecido pelos orientadores deste TFC.

Para alojar os algoritmos de *machine learning*, foi desenvolvido um servidor em Flask separado do servidor principal em Flask do iEquus, conforme abordado no capítulo 5. Este novo servidor Flask, que interage com os modelos de *machine learning*, dispõe de um *endpoint* funcional que recebe as coordenadas da imagem e devolve dois valores: BW e BCS. Foi também criado um *endpoint* adicional, invocado pelo servidor principal do *back-end*, para verificar a comunicação entre ambos.

F6

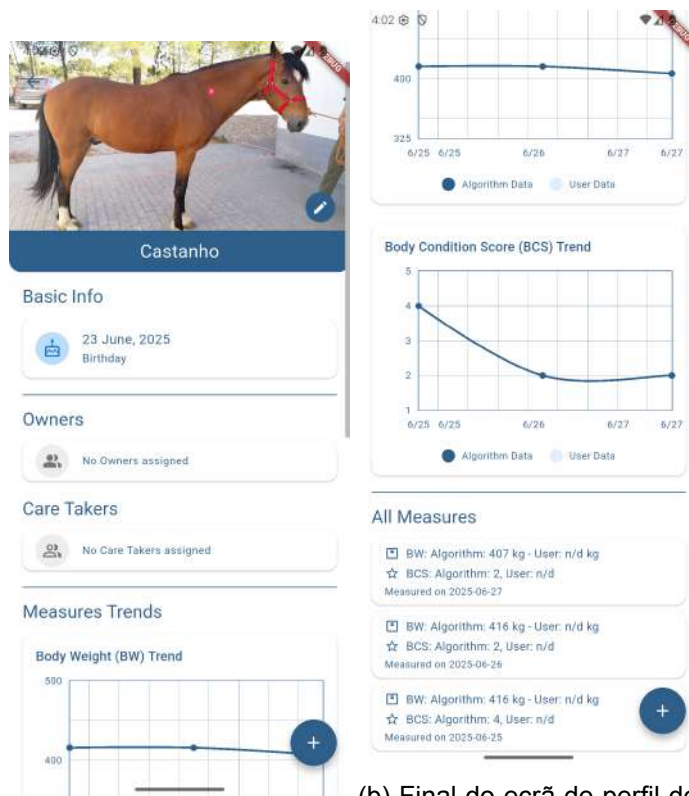
Para este requisito foi implementado o ecrã de registo e houve melhorias no ecrã de login, presente nas figuras 19, sendo que desta forma fica a faltar o ecrã para recuperação e alteração da password, apesar de que esta funcionalidade está totalmente preparada no *back-end* para implementação no *front-end*.

F7

Para finalizar este requisito, adicionaram-se as legendas do raio-X à imagem. Estas legendas, bem como as suas coordenadas, são enviadas do *back-end* para o *front-end*, que por sua vez as mapeia no ecrã. Ao clicar numa das legendas, como ilustrado na Figura 20a, é apresentado um *widget pop-up* com o nome do osso correspondente e uma breve descrição.

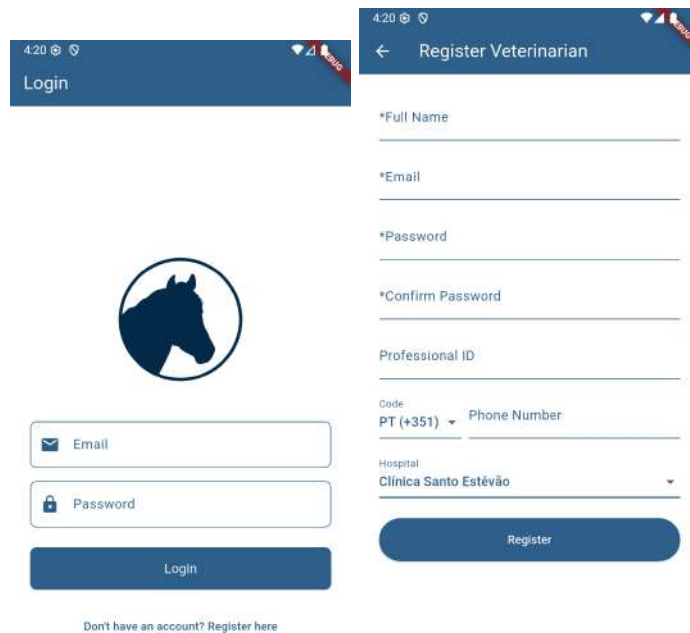
Do lado do *back-end*, foi implementado o *endpoint* responsável por fornecer estes dados. Embora ainda esteja em desenvolvimento, por parte dos orientadores deste TFC, um modelo de inteligência artificial capaz de reconhecer automaticamente a zona anatómica do cavalo presente na imagem, neste momento o servidor fornece apenas a imagem, como apresentado na Figura 20, e as legendas com as respetivas coordenadas.

8.2.2 Screenshots do front-end



(a) Início do ecrã de perfil de cavalos com o histórico de cavalos
(b) Final do ecrã de perfil de cavalos com o histórico de medições ao BW e BCS

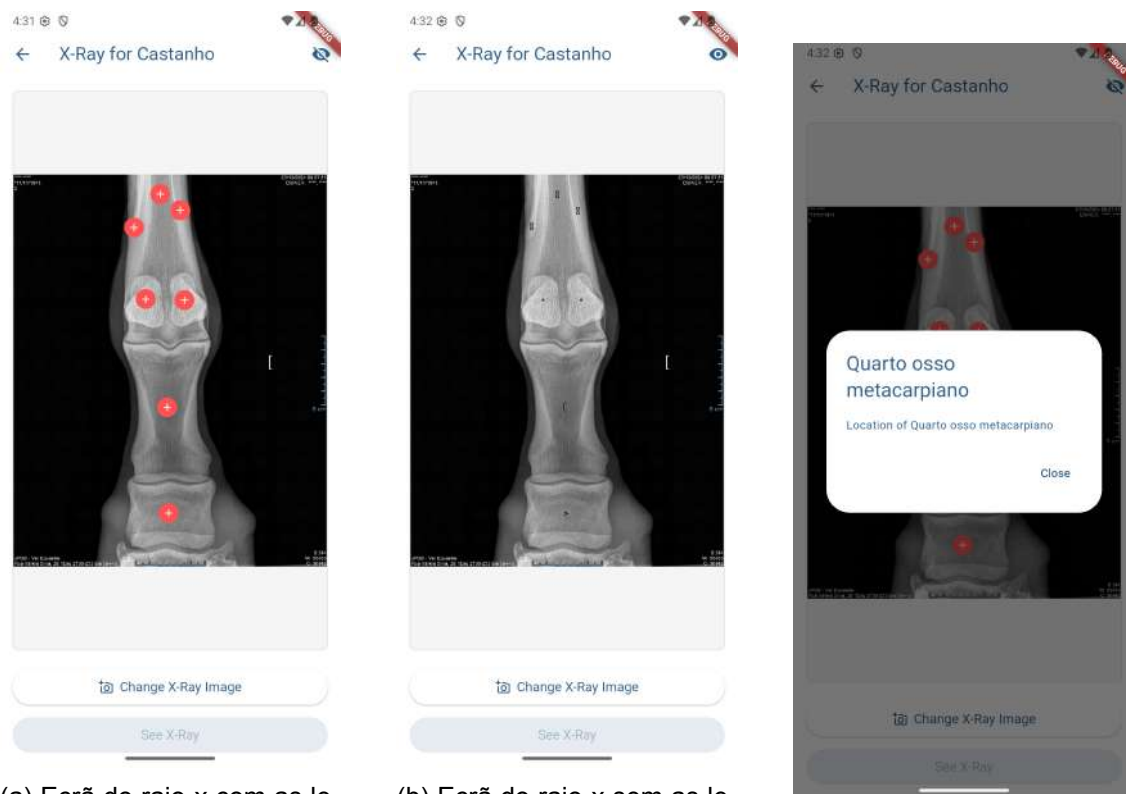
Figura 18: Screenshots do ecrã de perfil de cavalos



(a) Ecrã de *login*

(b) Ecrã de registo de um novo utilizador

Figura 19: *Screenshots* dos ecrãs de *login* e registo



(a) Ecrã do raio-x com as legendas marcadas

(b) Ecrã do raio-x sem as legendas marcadas

(c) Uma legenda da imagem

Figura 20: *Screenshots* dos ecrãs de *login* e registo

9 - Conclusão

O desenvolvimento do projeto iEquus progrediu significativamente em múltiplas áreas.

Primeiramente, o *benchmarking* permitiu identificar diversas aplicações relacionadas com a saúde animal já existentes no mercado, evidenciando a inexistência de soluções que integrem as funcionalidades específicas propostas neste projeto. A análise realizada revelou-se fundamental para compreender as lacunas e determinar os elementos a incorporar na aplicação, garantindo a sua diferenciação no mercado.

Adicionalmente, as entrevistas conduzidas com o Doutor José Prazeres, médico veterinário e docente, foram essenciais para identificar e validar os requisitos funcionais, bem como para adicionar novas funcionalidades e redefinir prioridades de desenvolvimento. Numa primeira fase, foi possível compreender as necessidades e desafios enfrentados no contexto prático da gestão da saúde equina, garantindo a adequação da solução proposta às necessidades reais, alinhada com as expectativas e exigências do setor.

Noutro plano, foi concluída a modelação da arquitetura do sistema, detalhando o funcionamento integrado das suas componentes, e desenvolvida a modelação relacional da base de dados, responsável por estruturar as entidades e relações fundamentais para a gestão e armazenamento da informação.

Paralelamente, iniciou-se o desenvolvimento da aplicação em Flutter, incluindo a configuração dos ambientes de trabalho, prosseguindo-se com a criação de protótipos para o desenvolvimento e validação das interfaces do utilizador.

Durante o processo, foram enfrentados desafios, nomeadamente ao nível das necessidades de aprendizagem da *framework* Flutter. A aquisição de competências nesta tecnologia decorreu em simultâneo com a sua aplicação prática, exigindo uma curva de aprendizagem acelerada.

Na segunda fase, foi possível avançar nos protótipos da aplicação, assim como na conclusão parcial do desenvolvimento do *back-end* e na implementação das principais funcionalidades do *front-end*.

Na terceira fase de desenvolvimento, concluíram-se os últimos passos de implementação do *front-end* da aplicação em Flutter, bem como o alojamento do *back-end* em Flask num servidor que, embora ainda não esteja nos servidores da Universidade Lusófona, encontra-se alojado numa solução doméstica. Ainda assim, esta versão do iEquus encontra-se operacional e apta a ser utilizada e testada pela Faculdade de Medicina Veterinária da Universidade Lusófona e pela clínica de Santo Estêvão.

O TFC culmina na realização dos testes funcionais em curso, de forma a garantir o alinhamento entre os requisitos técnicos já demonstrados em reunião e a validação da aplicação junto dos utilizadores finais, assegurando que as suas expectativas são plenamente atendidas.

Trabalhos futuros

Em termos de continuidade e perspetivas futuras, este TFC representa um ponto de partida com várias ferramentas integradas, mas que mantém ainda um vasto potencial de desenvolvimento adicional. Entre as possibilidades está a exploração da identificação de cavalos através da análise da castanha dos membros, uma característica distinta que pode ser aprofundada como técnica complementar de reconhecimento individual do cavalo e já preparada para desenvolvimento.

Outra vertente promissora prende-se com a integração de modelos de *machine learning* para detetar automaticamente as diferentes partes do corpo do cavalo, com vista à

criação de uma representação visual semelhante a um raio-x. Esta abordagem permitiria ampliar a utilidade prática da aplicação, tornando-a mais robusta e precisa na análise morfológica dos animais.

Paralelamente, ficou por explorar a opção de alojar o servidor deste trabalho nos servidores da Universidade Lusófona, o que poderia melhorar a escalabilidade e a segurança da aplicação.

Por fim, no que concerne o plano do desenvolvimento técnico, a aplicação poderá beneficiar de uma conclusão ao nível do *front-end*, nomeadamente com a implementação em Flutter. Esta fase torna-se particularmente viável, uma vez que o *back-end* já está completo, com todos os *endpoints* preparados para suportar o funcionamento integral da aplicação. Assim, abre-se caminho para uma solução final mais coesa.

Em suma, este trabalho não é estanque, pelo contrário representa um ponto de partida que poderá dar origem a futuros estudos e desenvolvimentos. A sua estrutura flexível e as várias possibilidades de expansão tecnológica permitem que novas abordagens venham complementar e aprofundar o que foi até agora desenvolvido, motivando a inovação contínua no que toca a ciência e tecnologia aplicadas ao contexto equino.

Bibliografia

- [Mat24] João P. Matos-Carvalho. *The Lusófona L^AT_EX Template User's Manual*. Lusófona University. 2024. URL: <https://github.com/jpmcarvalho/UL-Thesis>.
- [Gat] Bill Gates. *The road ahead reaches a turning point in 2024*. URL: <https://www.gatesnotes.com/The-Year-Ahead-2024> (acedido em 10/2024).
- [UZ23] Nadine Urbanek e Qendrim Zebeli. "Morphometric Measurements and Muscle Atrophy Scoring as a Tool to Predict Body Weight and Condition of Horses". Em: *Veterinary Sciences* 10.8 (2023). ISSN: 2306-7381. DOI: 10.3390/vetsci10080515. URL: <https://www.mdpi.com/2306-7381/10/8/515>.
- [HEN+83] D. R. HENNEKE et al. "Relationship between condition score, physical measurements and body fat percentage in mares". Em: *Equine Veterinary Journal* 15.4 (1983), pp. 371–372. DOI: <https://doi.org/10.1111/j.2042-3306.1983.tb01826.x>. eprint: <https://beva.onlinelibrary.wiley.com/doi/pdf/10.1111/j.2042-3306.1983.tb01826.x>. URL: <https://beva.onlinelibrary.wiley.com/doi/abs/10.1111/j.2042-3306.1983.tb01826.x>.
- [Bar] Lda. Barkyn. *Barkyn Assistant*. URL: <https://apps.apple.com/pt/app/barkyn-assistant/id6479705122?l=en-GB> (acedido em 11/2024).
- [Gla] University of Glasgow. *Equine Weight Management*. URL: <https://equibcs.en.aptoide.com/app> (acedido em 11/2024).
- [Ltd] EquiTrace Ltd. *EquiTrace*. URL: <https://equitrace.app/> (acedido em 11/2024).
- [AB] Sleip AI AB. *EquiTrace*. URL: <https://sleip.com/> (acedido em 11/2024).
- [Gmb] Siwalu Software GmbH. *Horse Scanner*. URL: <https://play.google.com/store/apps/details?id=com.siwalusoftware.horsescanner> (acedido em 11/2024).
- [San+24] Rafael Dos Santos et al. "Horses Identification Through Deep Learning Algorithms". Em: *2024 8th International Young Engineers Forum on Electrical and Computer Engineering (YEF-ECE)*. 2024, pp. 14–19. DOI: 10.1109/YEF-ECE62614.2024.10624972.
- [Enc] Let's Encrypt. *Let's Encrypt*. URL: <https://letsencrypt.org> (acedido em 06/2025).
- [ame] amen.pt. *amen.pt*. URL: <https://www.amen.pt> (acedido em 06/2025).

Anexos

9.1 Questionário de Satisfação

iEquus - Formulário de avaliação e sugestão

Olá, o meu nome é Duarte Chen e agradeço-te desde já por testares esta versão em desenvolvimento da aplicação iEquus.

Esta aplicação está a ser desenvolvida no âmbito do meu **Trabalho Final de Curso (TFC)** da Licenciatura em Engenharia Informática. Encontro-me atualmente numa fase decisiva do projeto, em que a recolha de **feedback real** é essencial para orientar os próximos passos e assegurar a qualidade da solução final.

Este formulário tem como objetivo reunir **opiniões construtivas** e **sugestões de melhoria** por parte de utilizadores que tiveram contacto com a aplicação — sejam colegas, docentes, profissionais na área veterinária ou cuidadores de cavalos.

É importante salientar que a aplicação ainda se encontra numa fase preliminar, com diversas funcionalidades por finalizar e um design ainda em evolução.

Justamente por isso, a tua opinião é especialmente valiosa neste momento.

📱 Podes descarregar a aplicação Android (APK) através do seguinte

link: https://drive.google.com/drive/folders/1Ym6yVOJGqupzk0SdjZjKx2a8hl65AaSq?usp=share_link

Agradeço o tempo disponibilizado e o contributo para o desenvolvimento do iEquus. A tua colaboração fará certamente a diferença na conclusão deste projeto.

1. Em que categoria te encaixas melhor? *

- ☐ Médico Veterinário
- ☐ Estudante de Medicina Veterinária
- ☐ Tratador de Equinos
- ☐

2. Registo | Conseguieste efetuar um registo numa conta pessoal? *

- ☐ Sim
- ☐ Sim, mas com dificuldade
- ☐ Não consegui

3. Login | Conseguieste fazer login na tua conta pessoal? *

- ☐ Sim
- ☐ Sim, mas com dificuldade
- ☐ Não consegui

4. Registo e Login | Se conseguiste fazer o registo ou o login, mas com dificuldade, ou não conseguiste, o que aconteceu?

5. Previsão do BW e BCS | Conseguiste tirar uma fotografia, seleccionar os pontos no cavalo e obter uma previsão do peso e da pontuação da condição corporal do cavalo? *

- ☐ Sim
- ☐ Sim, mas com dificuldade
- ☐ Não consegui

6. Previsão do BW e BCS | Se conseguiste, mas com dificuldade, ou não conseguiste, o que aconteceu?

7. Perfil de um cavalo | Conseguiste visualizar as informações de um cavalo? *

- ☐ Sim
- ☐ Sim, mas com dificuldade
- ☐ Não consegui

8. Perfil de um cavalo | O que mudarias neste ecrã?

9. Como descreves o aspeto visual da aplicação neste momento? *

- ☐ Nada apelativa
- ☐ Pouco apelativa
- ☐ Aceitável
- ☐ Boa
- ☐ Muito boa

10. Descreve a tua opinião sobre o atual aspeto visual da aplicação.

11. Foi fácil perceber como usar a aplicação? *

- ☐ Não percebi nada
- ☐ Foi confusa
- ☐ Percebi com alguma dificuldade
- ☐ Sim, foi claro
- ☐ Muito fácil e intuitivo

12. O que poderia ser mais fácil de usar na aplicação?

13. Como classificas a tua experiência geral com esta versão da aplicação? *



14. Queres partilhar mais alguma coisa?

This content is neither created nor endorsed by Microsoft. The data you submit will be sent to the form owner.

 Microsoft Forms

9.2 Manual de Instalação em Android

Manual de Instalação do iEquus (APK)

A aplicação está disponível em diferentes versões, dependendo do processador do seu telemóvel. Siga os passos abaixo para descobrir qual versão instalar:

1 Descobrir a Arquitetura do seu Telemóvel

Para saber qual é a arquitetura do seu processador (ARMv7, ARM64 ou x86_64), siga um destes métodos:

Método 1: Usar a App "Droid Hardware Info" (Recomendado)

- • Aceda à Google Play Store
- • Instale a aplicação [Informações do dispositivo](#) na PlayStore
- • Instale e abra a aplicação
- • Vá ao separador "CPU"
- • Veja o valor de "Supported ABIs"



Com base na arquitetura, instale o APK correspondente:

Valor encontrado	Arquivo que deve instalar
arm64-v8a / aarch64	✅ app-arm64-v8a-release.apk
armeabi-v7a / armv7	✅ app-armeabi-v7a-release.apk
x86_64	✅ app-x86_64-release.apk

Método 2: Verificar no Site da Marca

Se não quiser instalar uma app, procure no Google:

"[modelo do seu telemóvel] CPU architecture"

Exemplo:

"Samsung Galaxy A51 CPU architecture"

Depois use a tabela acima para saber qual APK instalar.

2 Instalar o APK

- Transfira o APK correto para o seu telemóvel.
- Se for necessário, ative a opção "Permitir instalação de apps de fontes desconhecidas" nas Definições ou no popup que surgir.
- Toque no ficheiro .apk para iniciar a instalação.
- Siga os passos no ecrã para completar a instalação.

! Notas Importantes

- Se instalar a versão errada, a app pode não abrir ou dar erro.
- Em caso de dúvida, a maioria dos telemóveis modernos é arm64-v8a.

Glossário

API *Application Programming Interface*. 6, 31, 33, 40, 41, 44

BCS *Body Condition Score - Pontuação da condição corporal*. 8

BW *Body Weight - Peso Corporal*. 8

CBC *Complete Blood Count*. 27

CNS *Cresty Neck Score*. 8

JWT *JSON Web Token*. 40, 41

MASS *Muscle Atrophy Scoring System*. 8

PDF *Portable Document Format*. 20, 21, 23, 27, 38, 40, 48

SWOT *Strengths, Weaknesses, Opportunities and Threats*. 9

TFC *Trabalho Final de Curso*. 11, 32–35, 39, 41, 46, 49, 52